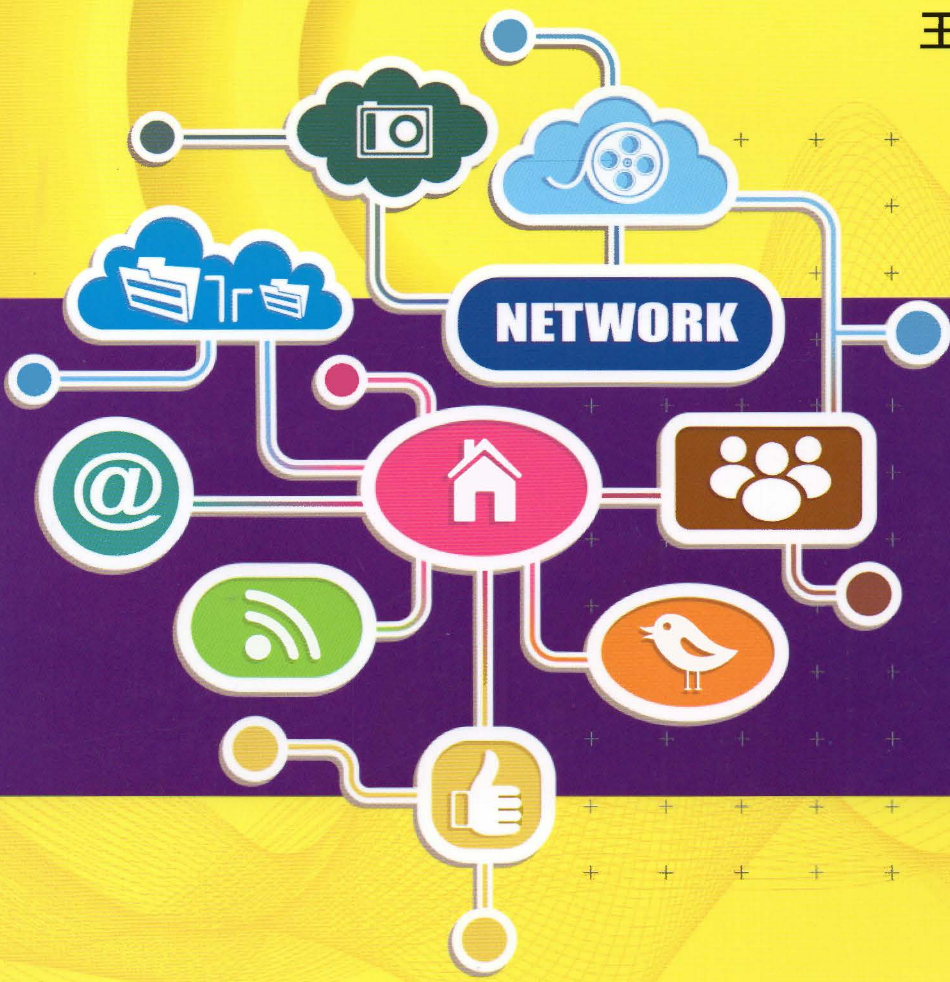
NFC技术原理与应用

王淼 等编著



化学工业出版社

NFC
技术原理与应用

王淼　等编著

果粉工作室
粉
RETRO WORKSHOP

化学工业出版社
·北京·

本书主要围绕 NFC Forum 的相关协议规范，全面地介绍了 NFC 技术原理、协议以及相关应用。本书共 14 章。第 1 章为概述，介绍了 NFC 的概念及其与 RFID 之间的联系与区别，详细介绍了 NFC 基本原理，如近场通信原理、负载调制、主被动通信等，并列举了 NFC 的相关应用领域。第 2 章为 NFC 协议概述，梳理了 NFC 相关的国际及行业标准情况，及其关联和演进关系，详细介绍了 NFC Forum 的协议规范组成及协议架构。第 3 章到第 10 章，根据 NFC 的三种工作模式（卡模拟、读写和点对点）详细介绍了 NFC Forum 协议架构中的每一个协议，包括 Analog、Digital/Activity、LLCP、NDEF、RTD、SNEP、标签操作规范和 NCI。第 11 章到第 14 章，针对目前 NFC 主要的应用领域，分别介绍了 NFC Forum 定义的辅助蓝牙/WiFi 连接切换规范、个人医疗健康设备通信规范和无线充电规范，并介绍了 NFC 在移动支付领域中的应用。

本书结构紧凑，内容由浅入深，主要面向 NFC 从业人员，如软件开发人员、科研人员、工程技术人员、通信相关专业的师生以及广大 NFC 技术爱好者。

图书在版编目（CIP）数据

NFC 技术原理与应用/王淼等编著．—北京：化学工业出版社，2014.5

ISBN 978-7-122-19983-6

Ⅰ．①N…　Ⅱ．①王…　Ⅲ．①超短波传播-无线电技术-研究　Ⅳ．①TN014

中国版本图书馆 CIP 数据核字（2014）第 043764 号

责任编辑：刘[illegible]青　　　　装帧设计：史利平

责任校对：吴

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号　邮政编码 100011）

印　　装：大厂聚鑫印刷有限责任公司

787mm×1092mm　1/16　印张 15½　字数 396 千字　2014 年 7 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899

网　　址：http://www.cip.com.cn

凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：58.00 元

版权所有　违者必究

FOREWORD 前言

NFC（Near Field Communication，近场通信）是一种工作频率为 13.56MHz，工作距离只有 0~ 20cm（实际产品大部分都在 10cm 以内）的近距离无线通信技术，允许电子设备通过简单触碰的方式完成信息交换及内容与服务的访问。

目前，NFC 技术已逐渐被应用到文件传输、移动支付、智能海报、公共交通、健康医疗等领域。特别是在移动支付领域，随着移动支付的兴起，央行在 2012 年发布了移动支付标准，NFC 成为非接触支付的技术标准。IDC 公司最新调查数据显示，至 2017 年，全球移动支付规模将超过 1 万亿美元，其中移动电子商务业务，包括通过手机的数字媒体消费以及浏览器页面网店购物等还将继续占据市场主要地位，市场份额紧随其后的业务将是借助 NFC 技术的交易。除了移动支付，NFC 还可以用于文件传输，如 Android Beam、三星 S Beam 等。在公共交通领域，NFC 手机作为公交卡、地铁票已经得到广泛应用，中国的很多城市陆续开通了该项业务。

NFC 技术从诞生至现在，技术已经成熟，正逐步形成一个完整的产业链，并涌现出大量有价值的应用和服务。随着智能手机的普及，越来越多的手机集成了 NFC 功能，NFC 技术将在各个领域发挥越来越大的作用。但是，目前关于 NFC 的书籍在国外尚不多，国内更是少见，而通信等相关专业的师生、科研人员以及业内人士都需要具备 NFC 基本工作原理和协议的知识，本书正是在这种情况下组织编写的。

本书介绍了 NFC 技术的基本原理，并围绕 NFC Forum 制定的协议架构，详细介绍了 Analog、Digital/Activity、LLCP、NDEF、RTD、SNEP、标签操作规范、NCI、连接切换、PHDC 及无线充电规范。在本书的最后一章，介绍了基于 NFC 技术的移动支付。

本书由王淼、曲文娟、曲文博、仇一平共同编著。同时要特别感谢王玥晗在编写过程中给予的宝贵意见及指导性建议。

由于编者水平有限，加上时间仓促，书中难免有不足之处，恳请广大读者批评指正。作者的电子邮箱为 NFC_Milton@ 163.com，欢迎与读者进行 NFC 技术的交流。

编　者
2014 年 1 月

Retro Workshop 果粉工作室

CONTENTS 目录

第1章 概述 …… 1

1.1 NFC技术的定义 …… 1

1.1.1 什么是NFC …… 1

1.1.2 什么是RFID …… 2

1.1.3 NFC与RFID的区别 …… 3

1.2 NFC的发展历史 …… 4

1.3 NFC通信基本原理 …… 8

1.3.1 近场通信原理 …… 8

1.3.2 NFC被动通信 …… 9

1.3.3 NFC主动通信 …… 9

1.3.4 负载调制 …… 9

1.4 NFC应用领域 …… 10

1.4.1 数据传输 …… 10

1.4.2 手机设置 …… 11

1.4.3 辅助WiFi/蓝牙配对 …… 11

1.4.4 移动支付 …… 13

1.4.5 智能海报 …… 14

1.4.6 航空 …… 15

1.4.7 医疗 …… 15

1.5 小结 …… 15

第2章 NFC协议概述 …… 16

2.1 NFC相关的标准化组织 …… 16

2.1.1 ISO …… 16

2.1.2 ECMA …… 17

2.1.3 NFC Forum …… 17

2.1.4 JIS …… 20

2.1.5 ETSI …… 20

2.1.6 NFC标准的演进 …… 21

2.2 NFC Forum协议架构 …… 22

2.2.1 卡模拟模式 …… 22

2.2.2 读写模式 …… 22

2.2.3 点对点模式 …… 23
2.3 小结 …… 23

第3章 Analog 规范 …… 24

3.1 Analog 规范概述 …… 24
3.1.1 NFC Forum 设备 …… 24
3.1.2 NFC Forum 参考设备 …… 25
3.1.3 工作空间 …… 26
3.2 射频功率与信号接口 …… 27
3.2.1 功率传输需求 …… 27
3.2.2 侦听设备对工作域的影响 …… 27
3.2.3 载波频率需求 …… 27
3.2.4 侦听设备的唤醒 …… 28
3.2.5 轮询设备射频冲突避免 …… 28
3.3 轮询设备到侦听设备的信号接口 …… 28
3.3.1 NFC-A 技术的调制需求 …… 28
3.3.2 NFC-B 技术的调制要求 …… 29
3.3.3 NFC-F 技术的调制要求 …… 30
3.4 侦听到轮询的信号接口 …… 31
3.4.1 侦听设备负载调制的通用要求 …… 31
3.4.2 NFC-A 副载波负载调制要求 …… 32
3.4.3 NFC-B 副载波负载调制要求 …… 32
3.4.4 NFC-F 副载波负载调制要求 …… 33
3.5 小结 …… 33

第4章 Digital/Activity 规范 …… 34

4.1 Digital/Activity 规范概述 …… 34
4.1.1 Digital 规范 …… 34
4.1.2 Activity 规范 …… 34
4.1.3 Digital/Activity 架构 …… 34
4.1.4 Digital/Activity 规范与 ISO 标准 …… 36
4.2 NFC-A …… 38
4.2.1 帧格式 …… 38
4.2.2 命令 …… 38
4.2.3 冲突检测 …… 44
4.3 NFC-B …… 49
4.3.1 帧格式 …… 49
4.3.2 命令 …… 49
4.4 NFC-F …… 52
4.4.1 帧格式 …… 52
4.4.2 命令 …… 52

4.4.3 冲突检测 …… 54
4.5 类型1标签平台 …… 55
4.5.1 封装格式 …… 55
4.5.2 命令 …… 56
4.5.3 冲突检测 …… 57
4.5.4 设备激活 …… 58
4.6 类型2标签平台 …… 58
4.6.1 封装格式 …… 58
4.6.2 命令 …… 58
4.6.3 设备激活 …… 58
4.7 类型3标签平台 …… 61
4.7.1 封装格式 …… 61
4.7.2 命令 …… 61
4.7.3 设备激活 …… 61
4.8 类型4A标签平台 …… 62
4.8.1 封装格式 …… 62
4.8.2 命令 …… 62
4.8.3 设备激活 …… 65
4.9 类型4B标签平台 …… 66
4.9.1 封装格式 …… 66
4.9.2 命令 …… 66
4.10 半双工通信 …… 67
4.11 ISO-DEP …… 67
4.11.1 封装格式 …… 67
4.11.2 数据交换 …… 68
4.11.3 设备去激活 …… 68
4.12 NFC-DEP …… 69
4.12.1 发起方和目标方 …… 69
4.12.2 封装格式 …… 69
4.12.3 命令集 …… 69
4.12.4 设备激活 …… 73
4.12.5 设备去激活 …… 76
4.13 小结 …… 76

第5章 LLCP规范 …… 77

5.1 LLCP概述 …… 77
5.1.1 LLCP协议架构 …… 77
5.1.2 LLCP链路服务类型 …… 78
5.1.3 无连接传输 …… 78
5.1.4 面向连接的传输 …… 78
5.1.5 SAP …… 79

5.2 LLCP 消息 …… 80
5.2.1 LLC PDU 格式 …… 80
5.2.2 LLC PDU 类型 …… 80
5.2.3 LLC PDU 参数 …… 85
5.3 LLC 工作流程 …… 87
5.3.1 链路激活 …… 87
5.3.2 链路去激活 …… 88
5.3.3 面向连接的流程 …… 88
5.3.4 无连接的流程 …… 90
5.3.5 链路维护流程 …… 90
5.3.6 服务发现流程 …… 90
5.4 MAC 层映射 …… 90
5.5 小结 …… 91

第6章 标签操作规范 …… 92

6.1 标签操作规范概述 …… 92
6.2 类型 1 标签操作规范 …… 92
6.2.1 存储结构和管理 …… 92
6.2.2 命令集 …… 96
6.2.3 NDEF 的发现及访问 …… 97
6.3 类型 2 标签操作规范 …… 99
6.3.1 存储结构和管理 …… 99
6.3.2 命令集 …… 100
6.3.3 NDEF 的发现及访问 …… 102
6.4 类型 3 标签操作规范 …… 102
6.4.1 存储结构和管理 …… 102
6.4.2 命令集 …… 104
6.4.3 NDEF 的发现及访问 …… 106
6.5 类型 4 标签操作规范 …… 108
6.5.1 存储结构和管理 …… 108
6.5.2 命令集 …… 109
6.5.3 NDEF 的发现及访问 …… 110
6.6 小结 …… 113

第7章 NDEF 规范 …… 114

7.1 NDEF 规范概述 …… 114
7.2 NDEF 消息格式 …… 114
7.3 Record 格式 …… 115
7.3.1 Record 结构 …… 115
7.3.2 MB …… 116
7.3.3 ME …… 116

7.3.4 CF …… 116
7.3.5 SR …… 116
7.3.6 IL …… 116
7.3.7 TNF …… 117
7.3.8 类型长度 …… 117
7.3.9 ID 长度 …… 117
7.3.10 负载长度 …… 117
7.3.11 类型 …… 117
7.3.12 ID …… 118
7.3.13 负载 …… 118
7.4 小结 …… 118

第8章 RTD 规范 …… 119

8.1 RTD 规范概述 …… 119
8.1.1 NFC Forum 已知类型 …… 119
8.1.2 NFC Forum 扩展类型 …… 120
8.2 文本 RTD 规范 …… 120
8.3 URI RTD 规范 …… 121
8.4 智能海报 RTD 规范 …… 124
8.5 签名 RTD 规范 …… 127
8.6 小结 …… 133

第9章 SNEP 规范 …… 134

9.1 SNEP 规范概述 …… 134
9.2 SNEP 消息 …… 135
9.2.1 SNEP 消息传输 …… 135
9.2.2 SNEP 请求消息 …… 135
9.2.3 SNEP 响应消息 …… 136
9.2.4 SNEP 消息的分段 …… 137
9.3 小结 …… 139

第10章 NCI 规范 …… 140

10.1 NCI 概述 …… 140
10.2 NCI 架构 …… 141
10.2.1 架构 …… 141
10.2.2 通信流程 …… 142
10.3 NCI 核心框架 …… 144
10.3.1 核心框架概述 …… 144
10.3.2 NCI 消息格式 …… 144
10.3.3 NCI 控制消息 …… 145

10.3.4 NCI 数据消息 …… 152
10.3.5 消息的分段与重组 …… 153
10.4 RF 发现过程 …… 154
10.4.1 RF 发现的参数配置 …… 154
10.4.2 开始 RF 发现过程 …… 157
10.4.3 选择目标 …… 161
10.4.4 RF 接口激活与去激活 …… 161
10.4.5 NFCEE 的 RF 发现请求 …… 163
10.4.6 RF 发现状态机 …… 164
10.5 RF 接口 …… 165
10.5.1 RF 接口架构 …… 165
10.5.2 Frame RF 接口 …… 168
10.5.3 ISO-DEP RF 接口 …… 170
10.5.4 NFC-DEP RF 接口 …… 171
10.5.5 NFCEE Direct RF 接口 …… 172
10.6 NFCEE 发现 …… 172
10.6.1 NFCEE 发现过程 …… 172
10.6.2 NFCEE 的开启和关闭 …… 174
10.7 NFCEE 接口 …… 174
10.7.1 NFCEE 接口架构 …… 174
10.7.2 APDU NFCEE 接口 …… 175
10.7.3 HCI NFCEE 接口 …… 175
10.7.4 支持类型 3 标签命令集 NFCEE 接口 …… 176
10.7.5 透明 NFCEE 接口 …… 177
10.8 参考附录表 …… 177
10.9 小结 …… 182

第11章 PHDC（个人健康设备通信）规范 …… 183

11.1 PHDC 概述 …… 183
11.2 点对点通信模式 …… 184
11.2.1 PHDC 点对点通信架构 …… 184
11.2.2 数据格式定义 …… 185
11.2.3 通信流程 …… 185
11.3 读写模式 …… 186
11.3.1 PHDC 读写模式通信架构 …… 186
11.3.2 数据格式定义 …… 187
11.3.3 通信流程 …… 187
11.4 小结 …… 188

第12章 Connection Handover 规范 …… 189

12.1 Connection Handover 概述 …… 189

12.2 连接切换流程……190
12.2.1 协商切换……190
12.2.2 静态切换……193
12.3 连接切换消息……193
12.3.1 消息格式……193
12.3.2 切换请求消息……194
12.3.3 切换选择消息……196
12.4 小结……197

第13章 NFC 无线充电技术……198

13.1 无线充电技术概述……198
13.1.1 无线充电发展历史……198
13.1.2 无线充电工作原理分类……201
13.1.3 无线充电标准介绍……204
13.2 NFC 无线充电系统架构……209
13.2.1 共用 NFC 天线的无线充电系统……209
13.2.2 NFC 协作式无线充电系统……210
13.3 NFC 无线充电控制协议……211
13.3.1 无线充电流程……211
13.3.2 控制流程……213
13.3.3 控制帧……215
13.3.4 控制流程的通信模式……216
13.4 小结……217

第14章 NFC 移动支付……218

14.1 移动支付概述……218
14.1.1 移动支付的概念……218
14.1.2 国内移动支付的发展……220
14.1.3 国外移动支付发展……222
14.2 NFC 移动支付技术……224
14.2.1 NFC 移动支付系统架构……224
14.2.2 SD 卡方案……225
14.2.3 嵌入式全终端方案……225
14.2.4 SWP SIM 卡方案……226
14.2.5 NFC 移动支付流程……229
14.3 NFC 移动支付应用分析……232
14.3.1 谷歌钱包……232
14.3.2 ISIS……234
14.4 小结……235

参考文献……236

第1章 概述

1.1 NFC技术的定义

1.1.1 什么是NFC

NFC（Near Field Communication，近场通信），是一种工作频率为13.56MHz，通信距离只有0～20cm（实际产品大部分都在10cm以内）的近距离无线通信技术。具有NFC功能的电子设备通过简单触碰的方式就可以完成信息交换及内容与服务的访问。

NFC具有三种工作模式：卡模拟模式、读写模式和点对点通信模式。

（1）卡模拟模式

将具有NFC功能的电子设备，如智能手机等，模拟成一张非接触智能（IC）卡，如银行卡、公交卡、门禁卡等。以NFC智能手机模拟成银行卡为例，用户只需要将该手机靠近银行POS机，就可以像使用银行卡一样完成支付交易。

（2）读写模式

具有NFC功能的电子设备，如智能手机，作为一个读卡器，可以读写IC卡、NFC标签以及工作在卡模拟模式下的NFC电子设备中的内容。例如，工作在读写模式下的电子设备充当一台POS机的角色，可以读取银行卡信息，从而完成金融交易。

（3）点对点通信模式

在两个具有NFC功能的电子设备间进行点对点通信，完成信息交换，如Android Beam中两部手机通过触碰，就完成了手机联系人信息或图片信息的交换。

基于以上三种工作模式，NFC被广泛地应用在门禁、公交车、手机支付、智能海报、数据传输等领域，如图1-1所示[1]。

用户可以将具有NFC功能的手机作为观看体育比赛的门票；从NFC智能海报上获取广告信息；通过NFC来帮助两个电子设备，如个人电脑和手机之间快速建立WiFi或蓝牙连接；将照相机贴近打印机，照片会自动地通过NFC传输给打印机完成打印；通过NFC手机刷卡乘坐地铁和公交车；通过两部NFC手机的触碰，完成名片、联系方式、图片、网页等数据交换；用户还可以使用NFC手机在商场、零售店等进行移动支付购物。

---

[1] NFC Forum. NFC Technology Overview. http：//www. nfc-forum. org/resources/presentations/.

实际上，NFC 的应用不仅仅局限上述生活领域，它还广泛应用于医疗、航空、汽车等领域。1.4 节将详细描述。



图 1-1　NFC 应用举例

NFC 具有两种通信方式：被动通信和主动通信。

（1）被动通信

被动通信方式是指由 NFC 通信的一方产生射频场，另一方从射频场中获取能量，并通过负载调制的方式与产生射频场的一方通信。被动通信方式经常用于卡模拟模式和读写模式中。这里需要强调一点，通信方式与 NFC 的工作模式没有必然的关系。在传统的读卡器和 IC 卡通信过程中，IC 卡是无源的，需要从读卡器产生的射频场中获取能量。因此当 NFC 设备模拟成一张卡的时候，绝大部分情况下都采用被动通信方式。NFC 设备一般情况下都是有源的，因此工作在卡模拟模式下的 NFC 设备是可以产生自己的射频场进行主动通信的，这种情况比较少见，但不代表卡模拟模式只能采用被动通信方式。

（2）主动通信

主动通信方式是指射频场由 NFC 通信双方交替产生，即通信双方在需要通信时产生自己的射频场，这就要求通信的双方都是有源设备。当通信的一方产生射频场进行通信时，另一方处于侦听模式，不会产生射频场。主动通信方式主要用于 NFC 点对点通信中。正如上面提到的，并不是点对点通信模式只能采用主动通信方式。关于主动通信和被动通信方式的详细内容将在 1.3 节介绍。

NFC 工作频率为 13.56MHz，通信距离为 0～20cm，目前有 106Kbps、212Kbps、424Kbps、848Kbps 四种数据传输速率。NFC 与其他无线通信技术在通信距离和传输速率上的区别如图 1-2 所示。

NFC 技术起源于 RFID 技术，因此想要了解 NFC 技术的历史、发展、演进，需要了解 RFID 技术。本节将详细说明 RFID 技术及 RFID 与 NFC 的区别。

1.1.2　什么是 RFID

RFID（Radio Frequency Identification）即射频识别技术，又称为电子标签、无线射频

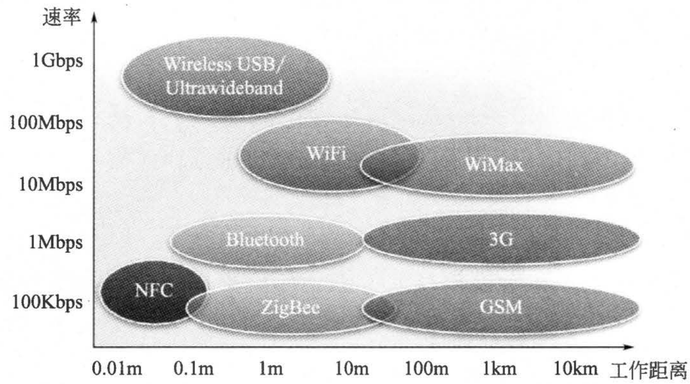


图 1-2 NFC 与其他无线通信技术的区别

识别，是一种射频通信技术，可通过无线电号识别特定目标并读写相关数据，无需识别系统与特定目标之间建立机械或光学接触。

目前 RFID 技术应用非常广泛，如图书馆、门禁系统、食品安全溯源、物流管理领域等。将 RFID 标签附着在一辆正在生产中的汽车上，厂方便可以追踪此车在生产线上的进度。将 RFID 标签附着在药品盒上，制药厂可以追踪药品的流向。标签也可以附着于牲畜与宠物上，方便对牲畜与宠物的识别。RFID 的身份识别卡可以作为门禁卡、工卡、身份证等用于确认持卡人身份。汽车上的射频应答器用于征收过路费与停车费。

RFID 的工作频率有低频（125～135kHz）、高频（13.56MHz）、超高频（860～960MHz）、微波等。RFID 在低频段的工作原理基于变压器耦合模型（初级线圈与次级线圈之间的能量传递及信号传递），在高频段基于雷达探测目标的空间耦合模型（雷达发射的电磁波信号遇到目标后，携带目标信息返回雷达接收机）。

RFID 系统由两部分组成，其工作原理如图 1-3 所示。

① 应答器：也称为电子标签，由天线、耦合元件及芯片组成。一般来说，都是用标签作为应答器。每个标签具有唯一的电子编码，附着在物体上标识目标对象。

② 阅读器：由天线、耦合元件和芯片组成，是读取（有时还可以写入）标签信息的设备，可设计为手持式 RFID 读写器或固定式读写器。

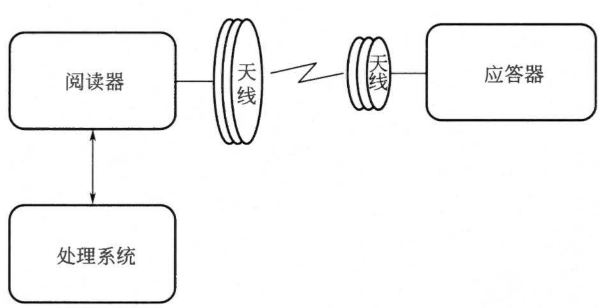


图 1-3 RFID 系统

在 RFID 系统中，电子标签附着在待识别物体的表面，电子标签中保存着约定格式的电子数据。阅读器可无接触地读取并识别电子标签中保存的电子数据，达到自动识别物体的目的。阅读器通过天线发送出一定频率的射频信号。当标签进入磁场时，产生感应电流，从而获得能量，发送出自身编码等信息；阅读器读取信息并解码后，送至电脑主机进行相关处理。

1.1.3 NFC 与 RFID 的区别

NFC 技术起源于 RFID，但是与 RFID 相比有一定的区别，主要包括以下内容。

（1）工作频率

NFC 的工作频率为 13.56MHz，而 RFID 的工作频率有低频、高频（13.56MHz）及超高频。

（2）工作距离

NFC 的工作距离理论上为 0～20cm，但是在产品实现上，由于采用了特殊功率抑制技术，使其工作距离只有 0～10cm，从而更好地保证业务的安全性。由于 RFID 具有不同的频率，其工作距离从几厘米到几十米不等。

（3）工作模式

NFC 同时支持读写模式和卡模拟模式。在 RFID 中，读卡器和非接触卡是独立的两个实体，不能相互切换。

（4）点对点通信

NFC 支持点对点通信模式，RFID 不支持点对点模式。

（5）应用领域

RFID 更多地被应用在生产、物流、跟踪、资产管理上，NFC 则在门禁、公交车、手机支付等领域发挥巨大的作用。

（6）标准协议

NFC 的底层通信技术兼容高频 RFID 技术（13.56MHz）的底层通信技术标准，即兼容 ISO 14443/ISO 15693 标准。NFC 技术还定义了比较完整的上层应用规范，如 LLCP、NDEF、SNEP、RTD 等。

尽管 NFC 与 RFID 技术有一些区别，但是 NFC 技术，特别是底层的通信技术是完全兼容 13.56MHz RFID 技术的。因此在 13.56MHz RFID 的应用领域中，同样可以使用 NFC 技术来代替。

1.2 NFC 的发展历史

NFC 起源于 RFID，因此介绍 NFC 的发展历史，要从 RFID 讲起。

RFID 技术最早追溯到第二次世界大战期间，它被用来在空战中进行敌我识别，雷达的发明和应用催生了 RFID 技术并奠定了 RFID 的理论基础。到了 20 世纪 70 年代末期，美国政府通过 Los Alamos 科学实验室将 RFID 技术转移到民间，最先商用在牲畜管理上。80 年代后，RFID 技术及产品进入商业应用阶段，各种规模应用开始出现。到了 90 年代，RFID 产品得到广泛应用，成为人们生活中的一部分。RFID 产品种类非常丰富，有源电子标签、无源电子标签、半有源电子标签发展迅速，成本不断降低，应用行业逐步扩大。

但是 RFID 的标准化问题日益严重。由于缺乏统一的标准，不同企业使用的 RFID 产品的频率、编码、存储规则以及数据内容等不尽相同，阅读器和标签不能通用，企业与企业之间无法顺利进行数据交换与协同工作，严重阻碍了 RFID 的发展。RFID 标准争夺的核心就是 RFID 数据内容的编码格式。国际上形成了五大标准组织，分别代表不同团体或者国家的利益。EPC Global 由北美 UCC 产品统一编码组织和欧洲 EAN 产品标准组织联合成立，在全球拥有上百家成员，得到了零售巨头沃尔玛，制造业巨头强生、宝洁等跨国公司的支持。AIM、ISO、UID 代表了欧美国家和日本；IP-X 的成员以非洲、大洋洲、亚洲等国家为主。比较而言，EPC Global 由于集合了美国和欧洲厂商，实力相对占上风。

在非接触卡的市场上，市场份额主要被两家电子巨头——飞利浦和索尼公司把持。飞利浦公司主导的 Mifare 最后成为 ISO 14443-A 标准。国内很多城市，如北京的公交卡系统就

采用了 Mifare 技术。国内也有一些兼容 Mifare 的产品，但是性能稍差。Mifare 的序列号全球唯一，不可修改，因此具有较高的安全性。索尼公司主推的是 Felica 技术。“Felica”是由英语单词幸福“felicity”和卡“card”组成。索尼公司想把 Felica 推成 ISO 标准，但是由于地域性及其他原因，并未被 ISO 接受。Felica 主要应用在日本市场，在香港的公交系统及国内少数城市如武汉的公交系统中也有使用。另外，由摩托罗拉公司主导的 ISO 1444-B 不如 Mifare 和 Felica 应用广泛，在欧洲一些国家如法国有一定的应用。由于标准不统一，各厂家各自为战，阻碍了市场的健康发展。因此，飞利浦公司和索尼公司整合了相关资源，共同推出了短距离通信技术和应用的统一方案，即现在的 NFC 技术。该技术融合了 Mifare 和 Felica，增加了点对点通信。

为了吸引更多的力量投入到 NFC 领域，加速 NFC 技术的普及，飞利浦、索尼和诺基亚公司牵头成立了 NFC Forum，开始推广 NFC 技术和商业应用。NFC Forum 的目的是开发 NFC 标准和互操作性协议，鼓励行业使用这些规范，从而形成一个完整的 NFC 生态系统。目前论坛已经有 12 位 Sponsor 成员，如图 1-4 所示；21 位 Principal 成员，其中包括中国的华为公司；78 位 Associate 成员，包括中国的中国移动、泰尔实验室、银联、中兴、联想等公司；另外还有大约 80 多位 Implementer 成员和 Non-Profit 成员，总计成员大约 200 位。

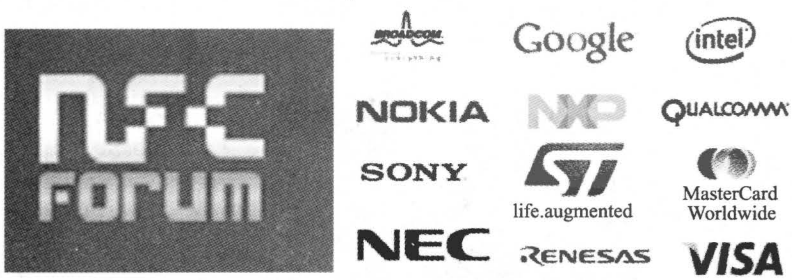


图 1-4　NFC Forum Sponsor 成员

从 RFID 的标准化历程及 NFC Forum 的成立来看，标准之争及市场新应用需求的出现是 NFC 出现并得以推广的原因。NFC 把 RFID 读卡器和智能卡功能结合在一起，可以直接利用现有的 RFID 基础设施。NFC 中加入了点对点通信，用于 NFC 设备之间的数据传输。

诺基亚作为 NFC Forum 的发起人之一，在 2004 年就推出了改版的诺基亚 6210 手机。该手机支持读写 RFID 标签，读取标签后可以触发短信程序发送短信。同年，诺基亚推出了 5140i 和 3220 手机的 NFC 外壳，如图 1-5 所示。安装了 NFC 外壳的 3220 手机，可以通过触碰 NFC 标签，访问网络和短信服务。该手机被德国的 RMV 公司应用到交通系统中提供交通卡服务，消费者可以将手机作为车票。在美国，3220 手机在 Visa、大通银行、ViVOtech 和 Cingular 公司的支持下在亚特兰大进行支付试点。

2006 年，诺基亚公司携手福建移动公司、厦门易通卡公司、飞利浦公司宣布，在厦门启动中国首个 NFC 手机支付现场试验。厦门移动公司招募百名志愿者，率先使用具有 NFC 功能的诺基亚 3220 手机，在厦门市任何一个厦门易通卡覆盖的公交汽车、轮渡、餐厅、电影院、便利店等营业网点进行手机支付，亲身体验 NFC 移动支付的便捷。用户只要刷一下手中的诺基亚 3220 手机，便可轻松实现各种易通卡交易，如图 1-6 所示。同年，诺基亚公司与银联合作在上海启动新的 NFC 测试。这是继厦门之后在中国的第二个 NFC 试点项目，也是全球范围首次进行 NFC 空中下载试验。

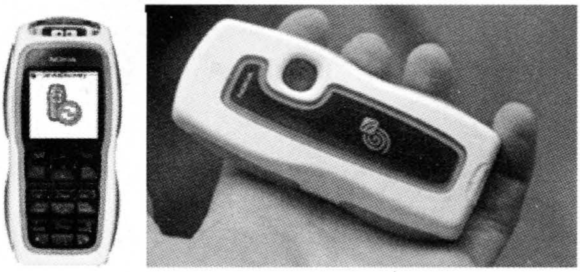


图 1-5　诺基亚 3220 NFC 外壳



图 1-6　厦门 NFC 手机易卡通

2007 年，诺基亚推出了 6131 手机。该手机采用内置 NFC 芯片，如图 1-7 所示，它不仅支持支付功能，而且可以通过智能海报获取商家电话并一键拨出，还可以通过与另一部 6131 手机触碰，交换各自的名片并存为联系人，通过触碰打印机自动触发蓝牙上传图片并打印。

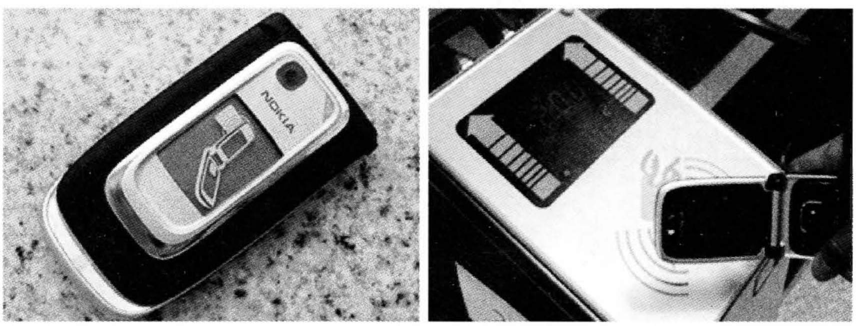


图 1-7　诺基亚 6131 手机

接下来，NFC 技术的发展一直不温不火。研究机构 ABI 在 2004 年预测，到 2009 年 50%的手机将具有 NFC 功能，但是实际上 1%都没有达到。以国内举例，尽管 2006 年在厦门和上海都开展了 NFC 支付的试点，但是由于当时 NFC 的生态环境尚不成熟，而且参与试点的各方之间的利益无法达成一致，最终两个试点项目都以失败而告终。手机支付可以说是 NFC 技术的一个杀手级应用，因此无论是运营商、银行，还是手机厂商、第三方支付公司，都积极地参与进来，但是利益如何分配、手机支付由谁来主导成为争夺的焦点，造成这个产业停滞不前，影响到 NFC 技术普及。另外，移动终端能力的限制、移动互联网基础设施的薄弱也是影响手机支付产业发展的因素。

当然这里有一个例外，就是日本市场。日本市场的手机几乎全由运营商定制及出售，因此运营商具备决定手机硬件支持 NFC 手机支付的能力。日本最大的运营商 NTT DoCoMo 入股了多家银行，为了降低更新 POS 终端对零售商带来的成本冲击，NTT DoCoMo 给予零售商一定的补贴来降低交易费用，推动了 NFC 手机的发展及手机支付的普及。图 1-8 所示为在日本通过刷手机来乘坐地铁。

业界有一个观点，就是 NFC 的惨淡发展或许与诺基亚公司有关。对于诺基亚公司来说，

在 2004 年到 2008 年这段时间，它在手机行业是当之无愧的“老大”，只有它才有能力推动 NFC 在手机上的集成。但是，诺基亚的 NFC 手机很少，而且都采用了 S30 或 S40 平台的低端产品，没有任何一款 Symbian 手机集成 NFC 功能；并且诺基亚的 NFC SDK（Software Development Kit，软件开发工具包）的授权需要 800 欧元，租用诺基亚服务器至少需要 400 欧元；另外，诺基亚公司对于开发者的支持也有限，所以试想一下，如果诺基亚公司能够全线推广 NFC 手机，并降低开发门槛，将自己的手机和各国的基础设施，如金融、公交等建立紧密的关系，也许今天的诺基亚是另一番景象[1]。业界还有一个观点，就是和恩智浦相关（2006 年，飞利浦公司的半导体部门独立出来，成立了恩智浦公司）。由于恩智浦公司是 NFC Forum 的主导者，而且其 Mifare 在全球占有绝对的市场，因此 NFC Forum 在成立后逐渐被恩智浦公司把持，这导致其他厂商参与的积极性下降。当然，这些观点不一定正确，但是从侧面说明了 NFC 发展不温不火的主要原因，那就是当时的 NFC 技术还不成熟，也没有一个盈利的商业模式；同时，NFC 产业链不完整，这才是 NFC 技术虽然出现很早，但是一直未得到很好发展的主要原因。

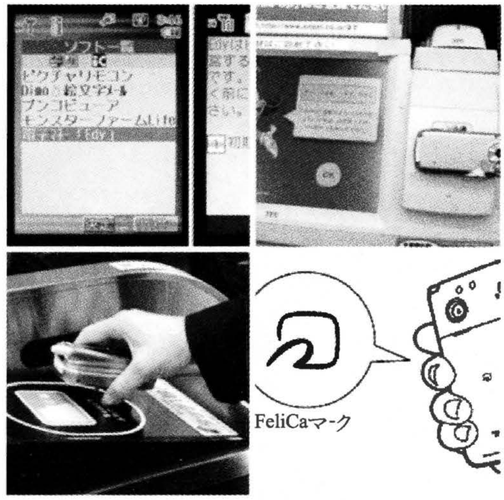


图 1-8 在日本通过刷手机乘坐地铁

随着智能手机的普及，谷歌 Android 的崛起，NFC 逐渐被各大手机厂商看好。谷歌公司率先发力，在 2011 年发布的 Android Gingerbread 系统中加入了 NFC 功能。NFC 协议栈由谷歌公司和恩智浦公司共同开发，在谷歌 Nexus S 上就采用了恩智浦的 PN544 NFC 芯片。Android NFC 协议栈完全免费。在应用上，谷歌公司推出了基于地理位置的服务 Google Hotspot。在 Google Place 注册的商家可以免费领取一个 Hotspot 的 NFC 标签，如图 1-9 所示，消费者使用 NFC 手机读取该标签，就可以在 Google Place 上实现对该商家进行评分、评论及签到的功能。

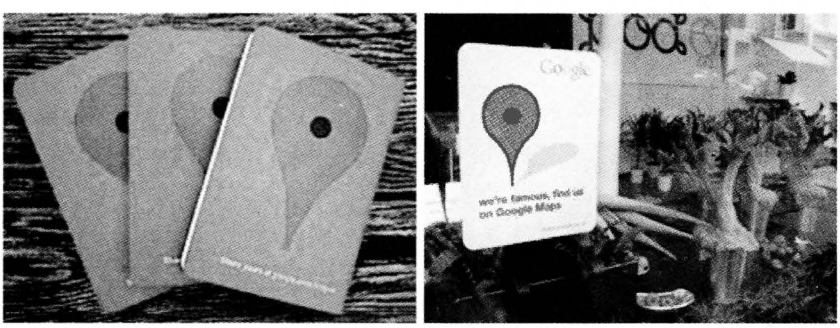


图 1-9 Google Hotspot NFC 标签

2011 年谷歌公司推出“谷歌钱包”，不仅展示了它的技术能力，还展示了其强大的商业

[1] 黄俊杰 . NFC 近距离无线通信技术的过去与未来 . http：//www. ifanr. com/27917.

资源整合能力。“谷歌钱包”不仅支持信用卡支付，还支持购物卡、礼品卡等支付方式。用户使用安装“谷歌钱包”的手机贴近具有 NFC 功能且支持“谷歌钱包”的 POS 机，就可以完成支付，如图 1-10 所示。



图 1-10　谷歌钱包

智能手机逐步普及，越来越多的手机集成了 NFC 功能，特别是主流厂商如三星、索尼、HTC、黑莓、诺基亚、联想、华为、中兴等的旗舰手机大都配置了 NFC 功能。另外，随着移动支付的兴起，央行在 2012 年发布了移动支付标准，NFC 成为移动支付非接触支付的唯一技术手段。除了移动支付，NFC 还可以用于快速文件传输，如在 Android 系统中的 Android Beam 及三星的 S Beam 都用到了 NFC 技术。此外，NFC 还在机场、公交、医疗、零售业等领域有着广阔的发展前景。如今，中国很多城市支持 NFC 手机刷公交卡。北京在 2013 年 9 月份正式启动。NFC 的标准化组织也在和航空业组织制定 NFC 在机场的应用标准，登机牌、护照等信息都可以存放在 NFC 手机中，用户只需要刷手机就完成登机手续。

IDC 公司最新调查数据显示，至 2017 年，全球移动支付规模将超过 1 万亿美元。其中，移动电子商务业务，包括通过手机的数字媒体消费以及浏览器页面网店购物等将继续占据市场主要地位，市场份额紧随其后的将是借助 NFC 技术的交易业务。

NFC 技术从诞生至现在，正逐步形成一个完整的产业链，并涌现出大量有价值的应用和服务。可以预见，NFC 将在社会生活各个领域发挥越来越大的作用。

1.3 NFC 通信基本原理

1.3.1 近场通信原理

对于天线产生的电磁场，根据其特性的不同，划分为三个不同的区域：感应近场、辐射近场和辐射远场。它们主要通过与天线的距离来区分。感应近场区指最靠近天线的区域。在此区域内，由于感应场分量占主导地位，其电场和磁场的时间相位差为 90°，电磁场的能量是振荡的，不产生辐射。辐射近场区介于感应近场区与辐射远场区之间。在此区域内，与距离的一次方、平方、立方成反比的场分量都占据一定的比例，天线方向图与离开天线的距离有关，也就是说，在不同的距离上计算出的天线方向图是有差别的。辐射近场区之外就是辐射远场区，它是天线实际使用的区域。在此区域，场的幅度与离开天线的距离成反比，且天线方向图与离开天线的距离无关，天线方向图的主瓣、副瓣和零点都已形成。

由于远场和近场的划分相对复杂，具体要根据不同的工作环境和测量目的来划分。一般而言，以场源为中心，在三个波长范围内的区域，可称为感应近场区；以场源为中心，半径

为三个波长之外的空间范围称为辐射场❶。

NFC 称为近场通信，其工作原理就是基于感应近场。在近场区域中，离天线或电磁辐射源越远，场强衰减越大，因此它非常适合短距离通信，特别是与安全相关的应用，如支付、门禁等。

1.3.2　NFC 被动通信

发起 NFC 通信的一方称为发起方，通信的接收方称为目标方。被动通信是指在整个通信的过程中，由发起方提供射频场，选择 106Kbps、212Kbps 或 424Kbps 其中一种速率发送数据；目标方不必产生射频场，而从发起方的射频场中获取能量，使用负载调制的方式，以相同的速率将数据回传给发起方，如图 1-11 所示。这里的目标方可以是有源设备，如处于卡模拟模式或点对点通信模式的智能手机，或者是无源标签，如 NFC 标签、RFID 标签等。本书中将通信的接收方，如有源设备和无源的标签，统一称为目标方。

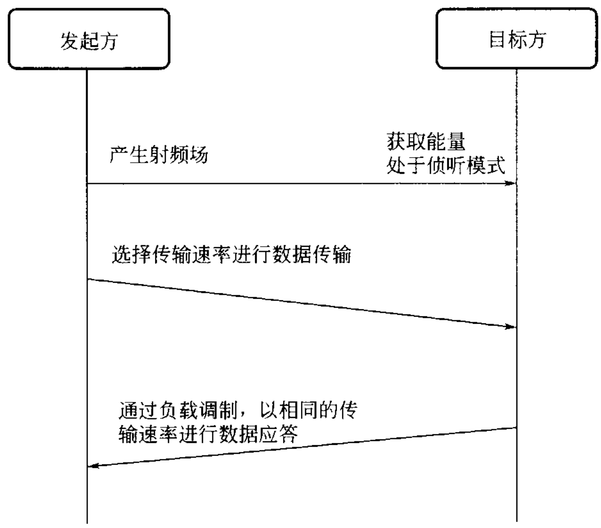


图 1-11　NFC 被动通信

1.3.3　NFC 主动通信

主动通信是指通信的发起方和目标方在进行数据传输时，都需要产生自己的射频场，如图 1-12 所示。当发起方发送数据时，它将产生自己的射频场，而目标方关闭射频场，并以侦听模式接收发起方的数据。当发起方发送完数据后，将关闭自己的射频场并处于侦听模式，等待目标方发送数据；目标方发送数据时，需要产生自己的射频场来发送数据。

主动通信要求通信的目标方是有源设备，即具有电源供给的设备。在通信过程中，发起方与目标方之间的关系是平等的，不存在主从关系，在发送数据的时候需要自己产生射频场；而另一方在没有数据发送或检测到周围空间有射频场的情况下，会关闭自己的射频场，在侦听模式下接收数据。因此，主动通信的方式一般适用于点对点的数据传输。

主动通信与被动通信相比较，由于主动通信的射频场分别由通信双方产生，因此在通信距离上比被动通信稍远。另外，在被动通信方式下，射频场由发起方提供，如果通信双方均为移动设备，将导致电源消耗不均衡，因此主动通信可以解决移动设备 NFC 通信过程中电源消耗的不平衡问题。

1.3.4　负载调制

在 NFC 被动通信中，通信的发起方产生射频场，而目标方通过负载调制将数据发送给发起方。

近距离通信系统的射频接口实际上是一个电感耦合系统，即一种变压器耦合系统。作为

❶ 电磁场的远场和近场划分. http://www.elecfans.com/tongxin/rf/20120829286379_4.html.

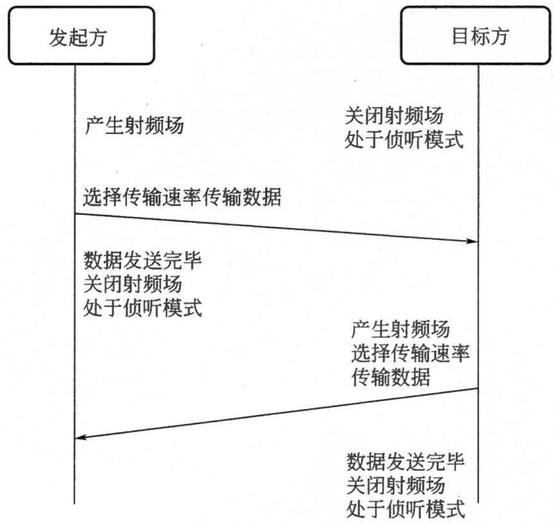


图 1-12 NFC 主动通信

初级线圈的发起方和作为次级线圈的目标方之间的耦合，只要线圈距离不大于 0.16 倍波长，该变压器耦合模型就是有效的。NFC 的工作频率为 13.56MHz，波长为 22m，因此只要 NFC 通信的发起方和目标方之间的距离不大于 3.52m，就遵循变压器耦合模型的定义。

如果目标方固有的谐振频率与发起方的发送频率相符合，那么把目标方放入发起方天线的交变磁场，目标方就能从磁场中获取能量。目标方天线的电阻成为发起方天线回路的负载。当负载电阻发生变化时，发起方天线的电流在内阻上的电压将变化。目标方通过待发送的数据控制负载电阻的接通和断开，可以实现目标方对发起方天线电压的振幅调制，数据就在 NFC 发起方和目标方之间传输。这种传输方式称为负载调制❶❷。

1.4 NFC应用领域

1.4.1 数据传输

尽管 NFC 的通信速率最高只有 848Kbps，但是与 WiFi、蓝牙等相比，NFC 连接建立的速度非常快，只需要通过触碰就可以完成数据传输，因此非常适合小数据的传输。

谷歌公司在 Android 4.0 系统中推出了基于 NFC 技术的 Android Beam，用于传输小量数据。只需要贴近两个支持 Android Beam 的手机，就可以传输如联系人、网页、照片等信息，如图 1-13 所示。举例来说，在一部智能手机上开启一个网页，之后贴近另一部支持 Android Beam 的手机，该网页将缩小，并提示用户触摸可发送。当用户触摸屏幕后，该网页将被传到另一部手机。Android Beam 采用 NFC 点对点通信方式进行数据传输。

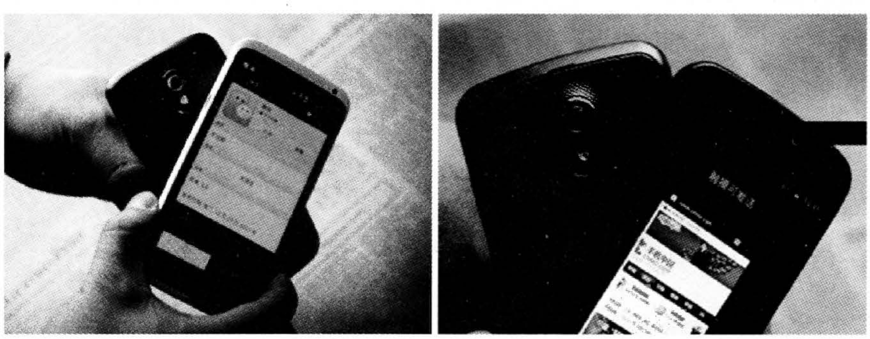


图 1-13 Android Beam

❶ Klause Finkenzeller. 射频识别（RFID）技术．电子工业出版社，2001.

❷ 曹平．高速率近距离通信（NFC）关键技术研究．硕士学位论文，2007.

另外，现在市面上很多文件管理软件都支持通过 NFC 传输文件，如 File Expert 新增了 NFC 文件传输功能。

1.4.2 手机设置

通过读取 NFC 标签，可以快速地设置手机。索尼移动、三星等手机厂商都推出了相关的应用。另外，在谷歌应用商店里可以看到很多类似的应用。本节着重介绍索尼移动公司推出的 Smarttags。

2011 年，索尼移动（索尼爱立信）公司发布的全线智能手机都具有 NFC 功能。为了推广 NFC 功能，索尼公司移动推出了 Smarttags 应用程序，通过读取 NFC 标签来快速设置手机，如图 1-14 所示。用户可以为每一个 Smarttags 标签设定应用场景，当手机贴近标签后，手机自动执行该应用场景的配置。

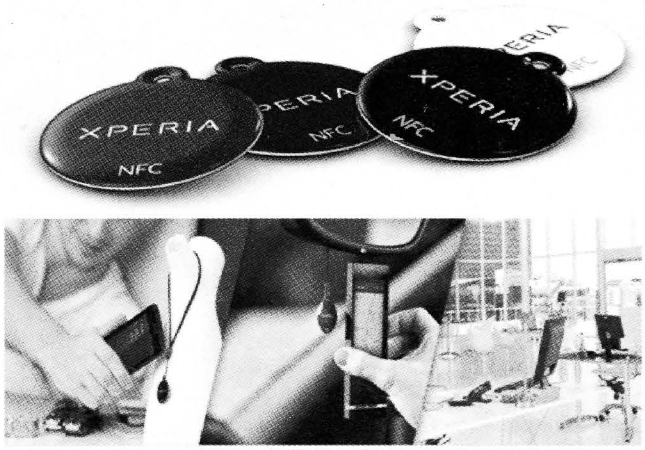


图 1-14 Smarttags NFC 标签

目前，Smarttags 推出了 4 种不同颜色的标签，用户可以为这些标签设定名称和要执行的动作。当用户第一次使用手机触碰这些标签时，Smarttags 应用程序识别出该标签，并显示在用户界面上。用户点击界面上的标签按钮，可以对其进行设置。例如，为红色的标签起名为“home”，用于家庭场景，然后通过“Add action”配置家庭场景，如打开 WiFi，将手机铃声音量调高，并启动“天气”和“新闻”的应用程序。设置完毕后，当用户回到家，用手机触碰该标签，手机的 WiFi 功能将会打开，音量调高，并启动“天气”和“新闻”应用程序，如图 1-15 所示。用户也可以设置其他应用场景，比如将蓝色标签设置为汽车场景，当用户进入车中，用手机刷一下挂在后视镜上的蓝色标签，手机的蓝牙程序将自动启动并连接到车载音响，或启动手机的 GPS 应用程序等。

三星公司也有类似的产品 Tectile，如图 1-16 所示。通过设定 Tectile NFC 标签，当手机靠近该标签后，所设定的配置将会在手机上执行。

1.4.3 辅助 WiFi/蓝牙配对

NFC 通过触碰就能快速完成数据交换的特性，被广泛应用到辅助 WiFi、蓝牙连接配对的场景中。由于 WiFi 和蓝牙搜索设备建立连接的速度比较慢，因此借助 NFC 交换 WiFi 或蓝牙配置信息，可以快速地完成 WiFi 或蓝牙的配对，从而建立连接。NFC Forum 也制定了 Connection Handover 规范来指导如何通过 NFC 为蓝牙、WiFi 等建立连接。

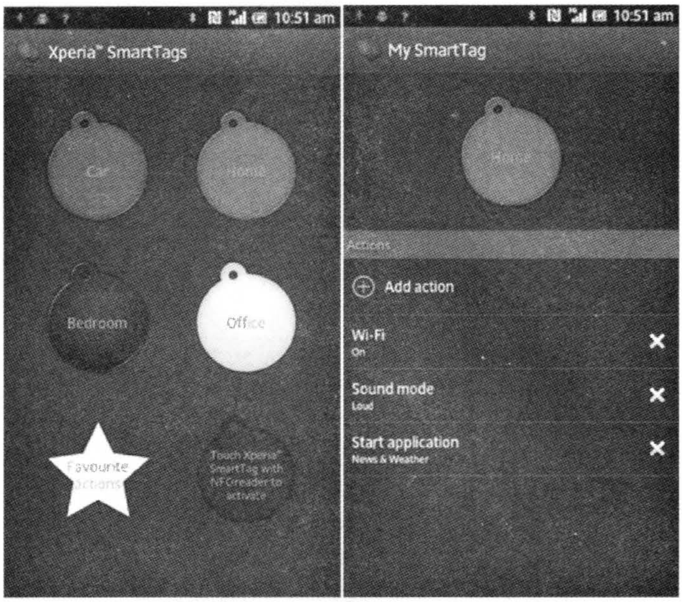


图 1-15 Smarttags 设置

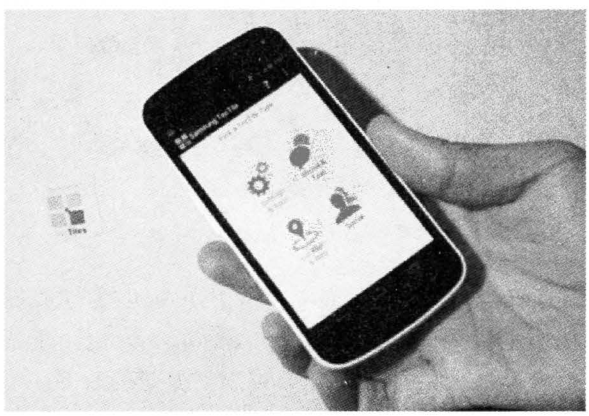


图 1-16 三星 Tectile

三星推出的 S Beam 就是通过 NFC 来辅助 WiFi Direct（WIFI 直连）建立连接的例子。尽管 Android Beam 实现了两部手机触碰后通过 NFC 进行数据传输的功能，但是只能传输一些较小的数据，比如图片、联系人信息、网页等，对于视频文件就无能为力了。三星在 Galaxy SⅢ上推出了 S Beam 的功能，当需要在两部手机之间传递视频或大容量数据时，只需要通过两部手机触碰就能在很短的时间完成数据传输，如图 1-17 所示。S Beam 的原理是这样的：两部手机触碰时，会通过 NFC 交换各自 WiFi Direct 的配置信息，免去了两部手机互相搜索对方并获取对方网络配置信息的过程，从而快速地建立 WiFi Direct 连接；然后，就可以通过 WiFi Direct 传输数据。总结一下，对于 S Beam，通过 NFC 来交换各自的 WiFi Direct 配置信息，从而建立 WiFi Direct 连接。视频等数据是通过 WiFi Direct 传输的。

其实在 S Beam 出现之前，NFC 就被很多手机厂商用在了辅助蓝牙配对上，比如诺基亚的手机与蓝牙耳机通过 NFC 蓝牙配对，如图 1-18 所示。蓝牙耳机具有 NFC 功能，与诺基亚 N9 触碰后，自动建立耳机与手机之间的蓝牙连接。



图 1-17 三星 S Beam



图 1-18 诺基亚蓝牙耳机与手机

此外，诺基亚公司有多款蓝牙耳机及便携式音箱配备了 NFC 功能。黑莓也有多款具有 NFC 功能的蓝牙耳机、蓝牙音箱和蓝牙多媒体网关，通过 NFC 完成手机和耳机、音箱、多媒体网关的蓝牙配对。

1.4.4 移动支付

正如前文介绍的，目前来说，移动支付是 NFC 技术最重要的一个应用。NFC 工作在 13.56MHz，与金融行业中非接触基础设施相兼容，并且由于工作距离非常近，具有天然的安全性，因此 NFC 也被作为移动支付非接触支付的技术标准。随着 NFC 智能手机的普及，各大运营商、银行、第三方支付公司、公交一卡通公司等都开始在移动支付领域发力。本节简单介绍移动支付相关应用，后续章节将详细介绍。

除了前面提到的“谷歌钱包”，还有美国三大运营商 AT&T、T-Mobile、Verizon 公司共同推出的 NFC 移动支付产品 ISIS，Visa 推出的 PayWave，MasterCard 推出的 Paypass，三星公司推出的“三星钱包”等，如图 1-19 所示。在国内，目前中国移动、联通、电信、银联等公司都推出了基于 NFC 技术的手机钱包业务。

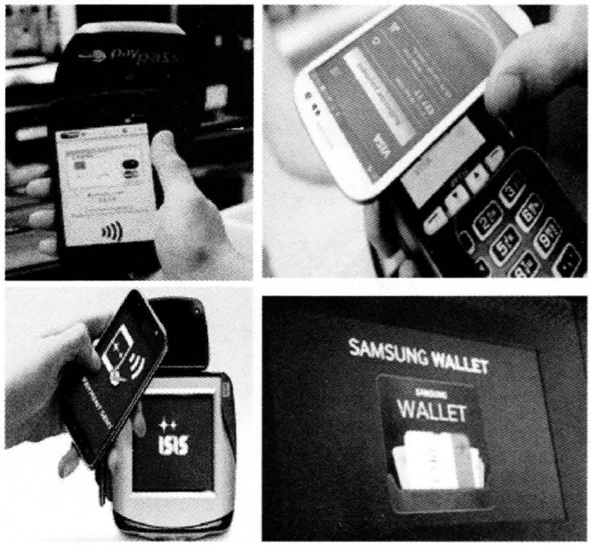


图 1-19 其他 NFC 移动支付应用

1.4.5 智能海报

智能海报是在原有纸质广告页、广告牌、商品包装、杂志等加入 NFC 标签，标签中存放如产品介绍、优惠券、票务等信息。如果 NFC 标签容量足够大，还可以存放产品的视频广告等多媒体信息。消费者只需将智能手机靠近智能海报的标签位置，就可以获得丰富的媒体内容，甚至开展互动。

NFC 智能海报在全世界很多地方都有应用，如图 1-20 所示。在我国台湾日月潭景区部署了智能海报，游客通过 NFC 手机触碰该海报，就可获取相关的景区信息，包括日月潭旅游地图。日本的 Strappy 系统在东京地铁的扶手柄内置了 NFC 标签，乘客可以使用 NFC 手机获取广告信息。著名的化妆品品牌雅诗兰黛为了纪念公司最热卖的产品，推出了智能海报，用户可以通过 NFC 手机访问该品牌的所有资讯。微软公司在澳大利亚推出了一项 NFC



图 1-20 NFC 智能海报

海报寻宝活动以推广“光晕 4”。其宣传海报上带有 NFC 标签以及二维码，而且本活动最大的特点是第一个读取 NFC 标签的玩家可以获得独一无二的奖励，后面的读取者只能获得基本的宣传内容。这种宣传方式可以激起玩家的寻宝式热情，全城搜索 NFC 广告，以获得第一次读取的奖励。三星公司与荷兰最大牛仔裤零售品牌 JeansOnline 合作，在荷兰阿姆斯特丹的 Kalverstraat 街区，消费者通过三星 NFC 手机扫描在商店橱窗广告中的 NFC 标签或者二维码，就可以获取商品信息，并获得代金券。

与二维码相比，NFC 智能海报的标签容量更大，可以存放丰富的媒体信息，而且标签中的内容可以更新。另外，用户只需触碰就可以获得信息，不像二维码那样需要启动摄像头并对准二维码，还会受到光线的影响等。但是，NFC 智能海报的成本比较高，这也是 NFC 智能海报未得到广泛应用的主要原因。

1.4.6 航空

NFC 技术不仅应用在公交、地铁系统中，还应用在航空领域。NFC Forum 正在与 IATA（国际航空运输协会）紧密合作，挖掘及评估 NFC 技术在航空运输中的应用场景。例如，用户可以通过手机从航空公司获取电子登机牌，在登机时只需要通过 NFC 手机刷卡就可以登机。另外，通过 NFC 手机在机场完成移动支付，从机场的智能海报获取航班信息、广告信息等，还可以通过 NFC 来认领行李等。

1.4.7 医疗

NFC 在医疗领域有非常丰富的应用场景。在 2012 年德国杜塞道夫医疗器材展及 2013 年年初美国消费性电子展（CES）中，日本医疗电子厂商如欧姆龙、索尼公司推出了以 NFC 为主要无线连接技术的居家照护产品，包含无线睡眠监控机、无线慢跑监控机、无线饮食监控机等；富士通公司也推出了具 NFC 功能的宠物计步器。

在医疗过程中，经常需要将患者当前的生理指标和状态结合历史治疗数据进行综合分析，以便医生做出准确判断和治疗。NFC 设备不仅能够实现设备之间数据的快速、安全传输，还能通过移动互联设备将实时采集的数据与患者历史治疗记录进行综合和智能分析，实现自动化、智能化的实时诊疗服务。例如，把 NFC 芯片内嵌到血糖仪或者创可贴等产品中，监视各项重要的人体生命特征，不仅了解病人之前的治疗情况，还可查询与病人相关的病例，促进医疗信息智能管理。因此，NFC 技术的应用将成为医疗行业的下一个爆发点[1]。

NFC Forum 与 Continua 健康联盟（Continua Health Alliance）紧密合作，发布了 PHDC（Personal Health Device Communication）规范。

1.5 小结

NFC 技术起源于 RFID，但是与 RFID 不同。本章首先介绍了 NFC 的概念及其与 RFID 之间的区别；然后介绍了 NFC 的基本原理，如近场通信原理、负载调制、主被动通信等；最后详细介绍了 NFC 的应用及应用领域。NFC 技术不仅用于数据传输、手机配置、辅助 WiFi/蓝牙配对、移动支付等场景，而且在医疗、航空、汽车等领域有着非常好的应用前景。

---

[1] NFC 技术应用将成医疗行业下一个爆发点 . http：//semi. cena. com. cn/2012-03/03/content _ 169570. htm.

第2章 NFC协议概述

2.1 NFC相关的标准化组织

2.1.1 ISO

国际标准化组织（International Organization for Standardization）简称ISO，是世界上最大的非政府性标准化机构，其任务是促进全球范围内的标准化及相关活动。各会员国以国家为单位，参加ISO技术委员会和分委员会的活动。通常情况下，ISO与IEC（国际电工委员会）合作制定全球性的国际标准，如图2-1所示。

图2-1 ISO/IEC标准化组织

在RFID及NFC领域，ISO/IEC制定了ISO 14443、ISO 15693、ISO 18092及ISO 21848等国际标准。

ISO 14443是非接触智能卡规范，即PICC（Proximity Integrated Circuit Card）规范。ISO 14443包括四个部分，第一部分规定了卡的物理特性，第二部分规定了射频功率及信号接口，第三部分规定了卡的初始化和防冲突特性，第四部分规定了卡片的数据传输协议。ISO 14443标准中的非接触式智能卡分为Type A和Type B两种类型。Type A是由飞利浦（恩智浦）等公司主导的产品，如Mifare系列产品。Type A技术和产品在世界上，特别是亚洲占据了很大的市场份额，被广泛应用于银行卡、公交卡、门禁等领域。Type B是由摩托罗拉等公司主导的产品，在世界上也有一定规模的应用。

ISO 15693也是非接触智能卡标准，即VICC（Vicinity Integrated Circuit Card）标准。ISO 15693包括四个部分，分别是物理特性、射频接口和初始化、防冲突和传输协议、扩展命令集和安全特性。ISO 15693的工作距离比ISO 14443远，但是目前市场上支持ISO 15693的产品较少，它主要应用于个人身份识别、会议签到、图书馆管理、门禁控制、物品跟踪、物品防伪、仓储物流等领域。

ISO 18092全称为Near Field Communication Interface and Protocol（NFCIP-1），即近距离通信接口和协议。该规范定义了工作频率为13.56MHz的电感耦合装置的近距离通信接口和协议，包括主动通信模式和被动通信模式、射频接口的调制方式、编码、传输速度和帧格式，以及在初始化过程中的防冲突机制、协议协商和数据交换的传输协议。

ISO 21481 全称为 Near Field Communication Interface and Protocol-2（NFCIP-2）。该规范的目的是将 ISO 14443、ISO 15693 以及 ISO 18092 融合在一起，通过一个标准的流程对这三种协议进行选择，使产品同时支持这三种协议并协同工作。

2.1.2 ECMA

ECMA（European Computer Manufacturers Association，欧洲计算机制造商协会）是一个国际性会员制度的信息和电信标准化组织，如图 2-2 所示。ECMA 的主要任务包括与其他标准化组织合作开发通信技术和消费电子标准，鼓励和落实标准文件与相关技术报告的出版。ECMA 组织出版的 370 项标准和 90 篇技术报告中，大约三分之二被 ISO/IEC 批准为国际标准。ECMA 是企业会员制组织，其标准化过程比较商业化。

图 2-2 ECMA 标准化组织

在 NFC 领域，ECMA 推出了 ECMA 340、ECMA 352 和 ECMA 385。其中，ECMA 340 称为 Near Field Communication Interface and Protocol（NFCIP-1），ECMA352 称为 Near Field Communication Interface and Protocol-2（NFCIP-2）。

正如前面提到的，ISO/IEC 是国际标准化组织，在全世界的认可度非常高，它经常与其他标准化组织合作完成标准化工作。ECMA 更多的是一个欧洲的标准化组织，影响力没有 ISO/IEC 大，因此 ECMA 的标准会在 ISO/IEC 中推动，使之成为 ISO/IEC 标准。ISO/IEC 和 ECMA 之间有一种称为“快速投票”的机制，允许 ECMA 标准快速地被 ISO/IEC 接受并成为 ISO/IEC 国际标准。ECMA 340 被 ISO 接受后成为 ISO 18092，ECMA 352 被 ISO 接受后成为 ISO 21481。

另外，ECMA 制定了 ECMA 385，被称为 NFC 的安全标准 NFC-SEC：NFCIP-1 Security Services and Protocol。该标准为 NFCIP-1 定义了安全通道和共享安全服务。ECMA 385 被 ISO 接受后成为 ISO/IEC 13157-1。

2.1.3 NFC Forum

在第 1 章中提到，NFC Forum 是由飞利浦公司、索尼公司和诺基亚公司牵头成立的，其主要工作是开发 NFC 标准和互操作性协议，鼓励行业使用这些规范，从而形成一个完整的 NFC 生态系统。

NFC Forum 的成员分为 5 类，即 Sponsor 成员、Principal 成员、Associate 成员、Non-Profit 成员和 Implementer 成员。

（1）Sponsor 成员

Sponsor 是最高等级的会员，拥有 NFC Forum 的董事会成员席位，目前有博通、诺基亚、三星、Intel、谷歌、高通、索尼、恩智浦、意法半导体、MasterCard、Visa、NEC 共 12 家成员。

（2）Principal 成员

Principal 是第二等级的成员，可以进入 NFC Forum 下属的各个委员会，并承担 NFC Forum 下属工作组主席职位，具有投票权。目前有 AT&T，NTT DoCoMo、黑莓、惠普、华为等 21 家成员。

（3）Associate 成员

Associate 成员可以参加 NFC Forum 委员会下属的工作组，但是没有提案权和投票权。

（4）Non-Profit 成员

Non-Profit 成员是一些非盈利组织，如 GSMA、CITC 等。

（5）Implementer 成员

Implementer 成员只能参加 SIG 组，只能获取技术委员会批准后的规范，无法获得标准、规范的讨论及修订细节。

董事会是 NFC Forum 的最高决策权中心，由 Sponsor 成员组成，主要对 NFC Forum 整体发展进行管理。董事会下属 4 个工作组和 3 个委员会，如图 2-3 所示。这 4 个工作组分别负责对外合作、组织流程管理、成员管理和日本市场 NFC 的推广管理工作。3 个技术委员会分别是 SIG 委员会、Compliance 委员会和技术委员会。

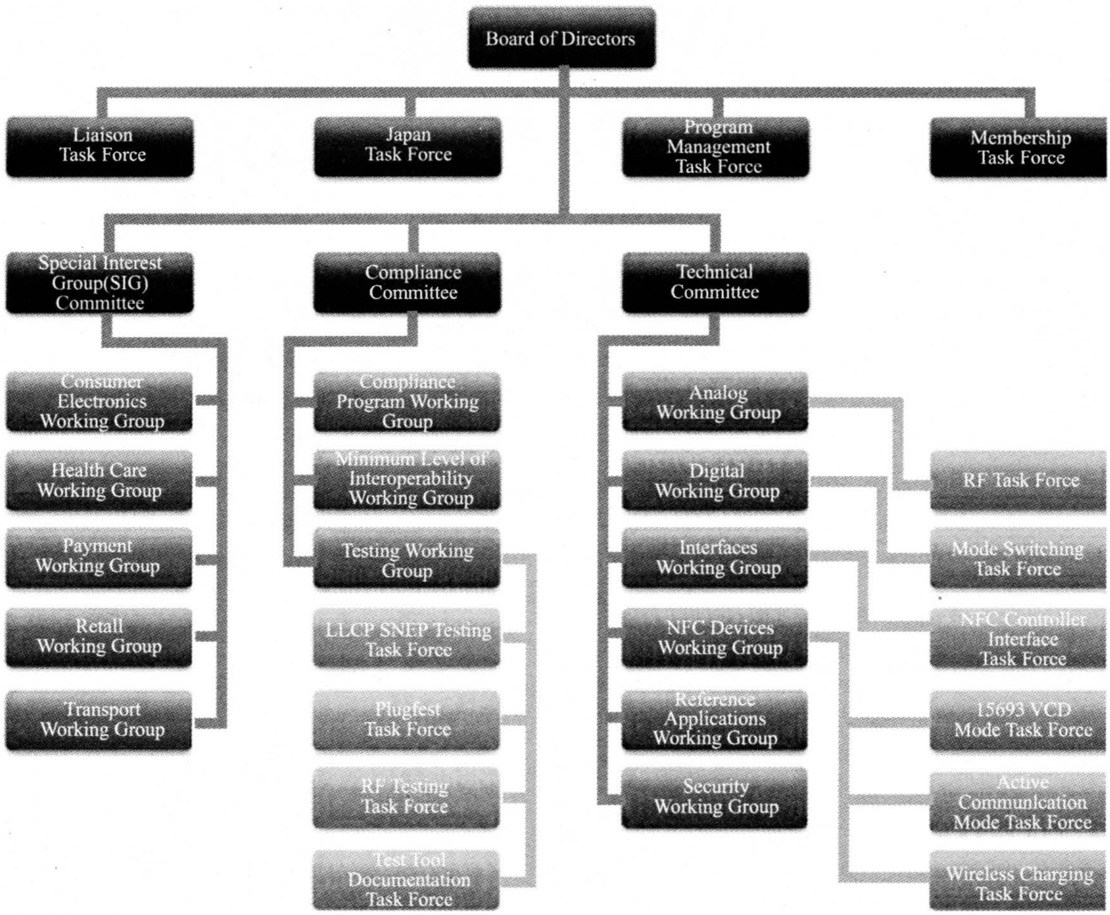


图 2-3　NFC Forum 组织架构

SIG 委员会主要负责收集各行业及真实生活中的 NFC 需求，并开发、挖掘 NFC 应用的商业机会。SIG 分为 5 个应用方向，分别是运输行业、健康、支付、零售和消费者电子。

Compliance 委员会负责保证 NFC Forum 发布的标准之间的一致性及互操作性。委员会下属 3 个工作组：MLOI、CPWG 和 TWG。

（1）MLOI（Minimum Level of Interoperability）

MLOI 工作组负责制定互操作性中哪些标准特性是必选的，哪些是可选的，并向 TWG 和 CPWG 提供这些信息。

（2）CPWG（Compliance Program Working Group）

CPWG工作组负责制定并管理认证流程，推动NFC相关的产品厂商来遵守测试规范及认证规范。

（3）TWG（Testing Working Group）

TWG工作组根据MLOI制定测试规范，根据CPWG定义的认证流程提供测试平台及工具。

技术委员会主要负责制定、维护NFC Forum的技术需求、技术规范，并为董事会和其他委员会提供技术支持，同时与其他组织进行技术合作。技术委员下属6个工作组，分别为Analog、Digital、Interface、Devices、Reference Applications（RAF）和Security。

（1）Analog工作组

负责制定并维护Analog规范。目前主要的工作是在Analog规范中支持ISO 15693。

（2）Digital工作组

负责制定并维护Digital规范。目前主要的工作是对现有规范中的一些问题进行修订。

（3）Interface工作组

负责制定并维护NCI规范，目前已经发布了1.0版本，正在根据新的需求制定1.1版本。

（4）Devices工作组

负责制定如LLCP、Type 1/2/3/4标签操作、无线充电等相关规范。

（5）Reference Applications（RAF）工作组

负责制定相关应用规范，如Connection Handover 1.3等。

（6）Security工作组

负责制定NFC相关的安全规范，如Signature RTD 2.0等。

NFC Forum不仅制定底层的通信标准（兼容ISO 14443/ISO 18092，目前正在兼容ISO 15693），而且针对卡模拟、读写、点对点这三种模式定义上层应用规范和接口规范。NFC Forum还制定了测试规范及认证规范，目的就是保证NFC产品之间互联互通。NFC Forum标准架构如图2-4所示。

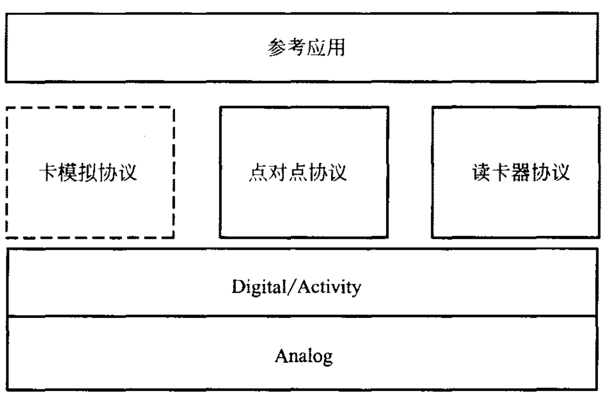


图2-4　NFC Forum标准架构

Analog规范主要定义NFC设备的RF特性，如波形、RF场强度，并定义RF的操作域，具体内容在第3章介绍。

Digital规范主要实现ISO 18092和ISO 14443中的数字部分，定义NFC基本的通信功能，如帧格式、编码、命令帧、数据帧等。Digital规范中定义了NFC通信的基本功能，Activity规范定义了如何使用Digital规范中的内容来建立通信。例如，在轮询模式下什么时

候执行冲突发现等。Digital 和 Activity 规范的具体内容在第 4 章介绍。

Analog、Digital 和 Activity 规范是 NFC Forum 标准架构的基础，NFC 的三种工作模式都基于这三个底层通信协议。在此之上，NFC Forum 针对不同的工作模式定义了不同的协议规范。此外，针对这三种工作模式制定了一些参考应用规范，即由 RAF 工作组完成。

NFC Forum 目前制定的技术标准如表 2-1 所示。

表 2-1 NFC Forum 标准

标准列表
Logical Link Control Protocol(LLCP)Technical Specification(1. 1)
Digital Protocol Technical Specification
Activity Technical Specification
Simple NDEF Exchange Protocol Specification
Analog Technical Specification
NFC Controller Interface(NCI)Technical Specification
NFC Data Exchange Format(NDEF)Technical Specification
NFC Forum Type 1 Tag Operation Specification 1. 1
NFC Forum Type 2 Tag Operation Specification 1. 1
NFC Forum Type 3 Tag Operation Specification 1. 1
NFC Forum Type 4 Tag Operation Specification 2. 0
NFC Record Type Definition(RTD)Technical Specification
NFC Text Record Type Definition(RTD)Technical Specification
NFC URI Record Type Definition(RTD)Technical Specification
NFC Smart Poster Record Type Definition(RTD)Technical Specification
NFC Signature Record Type Definition(RTD)Technical Specification
NFC Forum Connection Handover 1. 2 Technical Specification
NFC Forum Personal Health Device Communication Technical Specification

2.1.4 JIS

JIS（Japanese Industrial Standard）称为日本工业标准，是日本国家标准中最重要、最权威的标准。它由日本工业标准调查会（JISC）制定。FeliCa 是索尼公司推出的非接触式智能卡，在日本国内被广泛采用。可以说，Felica 在其国内的非接触卡市场中占有绝对的垄断地位，因此成为日本的国家标准 JIS X6319-4：High Speed Proximity Cards。由于索尼公司是 NFC Forum 的发起方之一，因此从开始就确定了 NFC 对于 Felica 的支持。

2.1.5 ETSI

ETSI（European Telecommunications Standards Institute，欧洲电信标准化协会）是由欧共体委员会 1988 年批准建立的一个非盈利性电信标准化组织，其总部设在法国南部的尼斯。ETSI 的标准化领域主要是电信业，涉及与其他组织合作的信息及广播技术领域。

NFC 可以被应用到各种垂直行业中，特别是移动支付中。移动支付对于安全性的要求比较高，因此在移动终端，如手机中的 NFC 芯片，不仅需要和手机的基带/应用处理器通信，还需要和安全芯片通信。安全芯片用于存储与金融相关的安全数据，如用户账号、密钥等信息。安全芯片可以集成在 SIM 卡中，也可以集成在 SD 卡中。因此，ETSI 制定了 SIM 卡与 NFC 芯片之间的接口规范 ETSI TS 102 613 和 ETSI TS 102 622。

ETSI TS 102 613 定义了 SWP（Single Wired Protocol，单线协议），用于 UICC（SIM 卡）和 CLF（NFC 芯片）之间的物理层和数据链路层协议。SWP 在一条单线上实现了全双工通信。ETSI TS 102 622 定义了 HCI（Host Control Interface，主机控制接口），用于主机之间的通信接口，在 SWP 协议之上负责 UICC 和 CLF 之间逻辑传输管道的建立和路由。关于 SWP 和 HCI 的详细内容，将在第 13 章详细介绍。

2.1.6 NFC 标准的演进

NFC 的相关标准众多，图 2-5 描述了 NFC 相关标准之间的关系，图中并未包含其对应的测试规范。

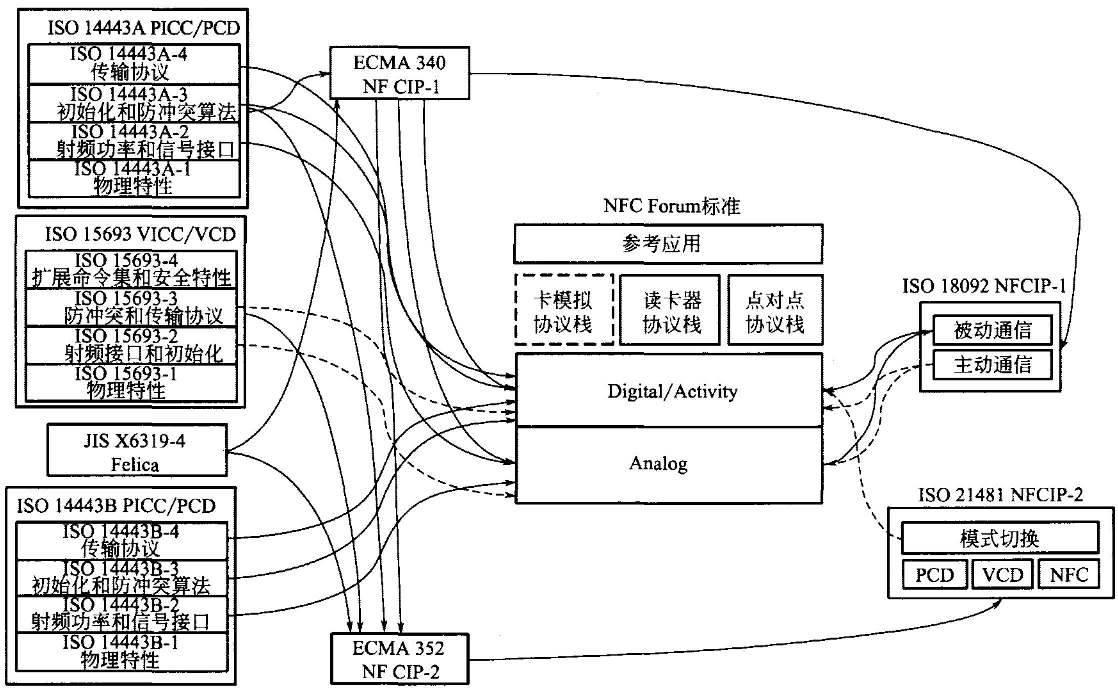


图 2-5　NFC 相关标准的关系

从图中可以看出，NFC Forum 与其他标准化组织的区别主要在于：NFC Forum 制定了一套支持点对点模式和读写模式的协议栈以及相关的上层参考应用，而在底层的规范完全纳入了 ISO 14443/ISO 15693/ISO 18092/ISO 21481/JIS X6319-4。对于 ISO 15693 和 ISO 21481，NFC Forum 正在把这两个标准融合进 NFC Forum 的 Digital 规范和 Analog 规范中，目前工作还在进行中，因此图中用虚线来表示。

关于 ECMA 340 和 ECMA 352，前面提到 ECMA 和 ISO 合作的机制，因此不难理解，ECMA 340 和 ECMA 352 分别对应着 ISO 18092 和 ISO 21481。在 ECMA 340 中，采用 ISO 14443A 和 JIS X6319-4 作为其通信基础，ECMA 340 中并没有采用 ISO 14443B 的部分。ECMA 340 之后被 ISO 接受，成为 ISO 18092。ECMA 352 的目的是将 ISO 14443A、ISO 15693 和 ISO 18092/ECMA 340 在模式选择的流程上统一起来，之后被 ISO 接受成为 ISO 21481。NFC Forum 技术委员会下属的 Digital 工作组中设有一个 Mode Switching 的子工作组，主要负责将 ISO 21481 集成进 Digital 规范和 Activity 规范，目前最终版本还未发布，因此这里用虚线代替。

目前来说，NFC 的标准主要由 NFC Forum 制定，但是 NFC Forum 其实是一个行业组织，它也必须遵循国际标准，特别是底层的通信标准。因此，NFC Forum 的 Digital、Ac-

tivity、Analog 这三层标准完全来自于 ISO 标准，而 NFC Forum 的主要工作是解决兼容性及互操作性的问题。

2.2 NFC Forum 协议架构

第 1 章介绍了 NFC 的三种工作模式，本节将介绍这三种模式所对应的标准协议架构。

2.2.1 卡模拟模式

在卡模拟模式下，NFC Forum 没有制定其他规范来支持该模式，如图 2-6 所示。ISO 14443 相关的内容都已经包含在 Digital/Activity 和 Analog 规范中。应用层一般由应用开发者完成，比如在智能手机上开发一个模拟卡的应用，其他手机靠近后可以读取其中的内容。NFC 协议栈为开发者提供了通信通道和读取命令，但是数据内容如何解析，用户界面如何设计，就不是 NFC 协议栈的任务。尽管 NFC Forum 制定了 NDEF（NFC Data Exchange Format）规范，但是在卡模拟模式的标准架构中没有给出倾向性的主要原因是目前市面上大部分 ISO 14443/15693 的基础设施还不支持 NDEF 格式，而且就目前而言，卡模拟模式的应用比点对点和读卡器模式更广泛。因此，为了兼容这些基础设施，NFC Forum 没有给建议，完全由应用开发或服务提供商来决定其数据格式。

2.2.2 读写模式

NFC Forum 制定了一套规范来支持读写模式，如图 2-7 所示。

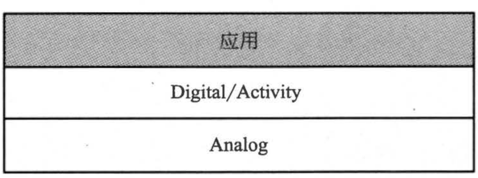


图 2-6 卡模拟标准架构

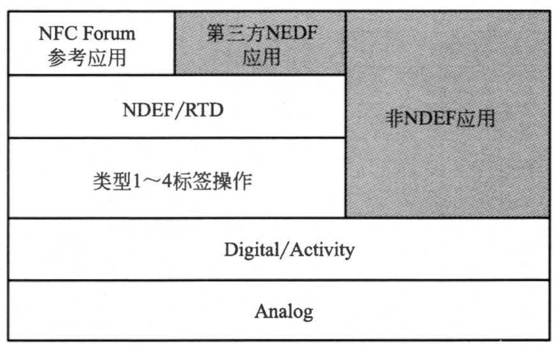


图 2-7 读卡器标准架构

NFC Forum 定义了四种类型的标签。

① 类型 1：该标签是基于 ISO 14443A 的私有标签。这里“私有”的意思是指该标签尽管基于 ISO 14443A，但是采用了私有的加密算法。类型 1 标签主要是与 Innovision Topaz 的产品兼容。

② 类型 2：该标签是基于 ISO 14443A 的私有标签，兼容的产品是恩智浦公司的 MIFARE Ultralight 和 MIFARE Ultralight C。

③ 类型 3：该标签兼容 Felica 产品。

④ 类型 4：该标签分为类型 4A 和类型 4B，分别兼容 ISO 14443A 和 ISO 14443B。

类型 1/2/3/4 Tag Operation 定义了如何从类型 1/2/3/4 的标签中读取或写入 NDEF 消息。相应的，在 Digital 规范中定义了与这四种标签进行通信操作的标签平台（Tag Platform）。

NEDF 定义了 NFC 设备之间或设备与标签之间通用的数据格式。NDEF 由 RTD

(Record Type Definition）组成。RTD 定义了不同数据类型的封装格式，如 URL、智能海报、文本等等。

在应用层，NFC Forum 制定了基于 NDEF 的参考应用，如 Connection Handover 中的静态切换。另外，第三方应用程序可以基于 NDEF 消息进行读写器应用的开发；同时，NFC 协议栈支持一些非 NDEF 的应用。

2.2.3　点对点模式

NFC Forum 点对点通信的标准架构如图 2-8 所示。

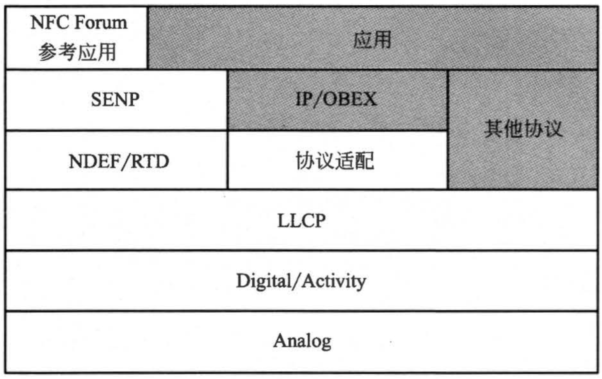


图 2-8　点对点标准架构

NFC Forum 定义了 LLCP（Logical Link Control Protocol）作为点对点通信的逻辑链路管理协议。LLCP 主要负责链路的激活、去激活和维护，为上层应用提供面向连接的服务和非连接服务，并提供协议复用和异步平衡模式等功能。

LLCP 协议层之上可以是基于 NDEF 的应用，也可以是在 NFC Forum 注册的协议，如 IP、OBEX。如果应用层基于这两种协议，那么 NFC Forum 针对这两个协议分别定义了一个适配协议，用于将 IP 和 OBEX 的协议数据映射进 LLCP 的帧格式。目前，除了这两个协议之外，如果应用开发者希望使用 LLCP 的功能，需要自己完成映射关系。目前 NFC Forum 只提供 IP 和 OBEX 的映射。

SNEP（Simple NEDF Exchange Protocol）称为简单 NDEF 交换协议，用于在点对点模式下两个 NFC 设备通过请求/响应的方式进行 NDEF 消息的交换。SNEP 是基于 LLCP 的面向连接服务。基于 SNEP，NFC Forum 也制定了相关的参考应用，如 Connection Handover。这里需要注意的是，NFC Forum 制定的参考应用规范，即 RAF 组的规范，没有强制力，是 NFC Forum 为了推广 NFC 应用而制定的一个参考性规范。上层应用可以基于 SNEP，也可以基于 IP 或 OBEX，或者直接基于 LLCP。

2.3　小结

NFC 的相关标准众多，本章首先梳理了 NFC 的标准化组织及其标准制定情况，总结了其他标准化组织及其标准与 NFC Forum 标准之间的关系；然后详细介绍了 NFC Forum 的协议架构，包括卡模拟、读写和点对点通信三种模式下的 NFC 协议栈，为读者完成后续章节的学习奠定基础。

第3章 Analog规范

3.1 Analog 规范概述

Analog 规范[1]定义了 NFC 设备射频的模拟特性。这里的 NFC 设备是指处于点对点模式下的发起方设备、点对点模式下的目标方设备、读写模式的设备和卡模拟模式的设备。这些设备可以工作在 NFC-A、NFC-B 和 NFC-F 三种技术下，以及 106Kbps/212Kbps/424Kbps 速率下（由于主动通信模式还未正式加入 Analog 规范，因此本节未列举 848Kbps）。NFC-A、NFC-B 和 NFC-F 将在第 4 章 Digital 规范中详细讲解，这里仅简单介绍。“技术”是 NFC 标准中的一组传输参数，例如射频载波、通信模式、比特率、调制方式、帧格式等。在 Digital 规范中定义了三种技术：NFC-A、NFC-B 和 NFC-F。简单来说，NFC-A 的参数兼容 ISO 18092 和 ISO 14443A，NFC-B 的参数兼容 ISO 14443B，NFC-F 的参数兼容 ISO 18092 和 JIS X 6319-4（Felica）。

Analog 规范没有给出天线的设计方案，它主要是通过一个外部的观测设备来评估 NFC 设备射频模拟特性参数，指导天线设计人员如何进行参数取值。这些特性包括功率需求、传输需求、接收需求和信号的形式（如时间、频率、调制特性等）。

3.1.1 NFC Forum 设备

Analog 规范专门介绍了 NFC Forum 设备，从而区别非 NFC Forum 的设备，主要原因是 Analog 规范描述的所有射频模拟特性都是基于 NFC Forum 设备的。本章为简便起见，统称为 NFC 设备，专指 NFC Forum 定义的设备。

Analog 规范定义的 NFC 设备具有射频天线，并具有侦听（Listening）和轮询（Polling）功能，其工作原理与变压器类似，如图 3-1 所示。

当可变的电流经过轮询设备的初级线圈后，将产生交变的电磁场。该电磁场会在侦听设备的次级线圈上产生电流。侦听设备通过这个射频场给自己的电路提供电能，从而激活自己。轮询设备是指处于轮询模式下的 NFC 设备，侦听设备是指处于侦听模式下的设备。轮询模式或侦听模式是 NFC 设备的初始状态。轮询模式是指 NFC 设备产生射频场并发送轮询命令，如探测周围设备的命令；侦听模式是指 NFC 设备不产生射频场，只是在侦听其他设备产生的射频场。

将数据承载在信号载波的过程称为调制，承载可以通过幅度、相位、频率三种方式来完

[1] NFC Forum. NFC Forum-TS-Analog-1. 0.

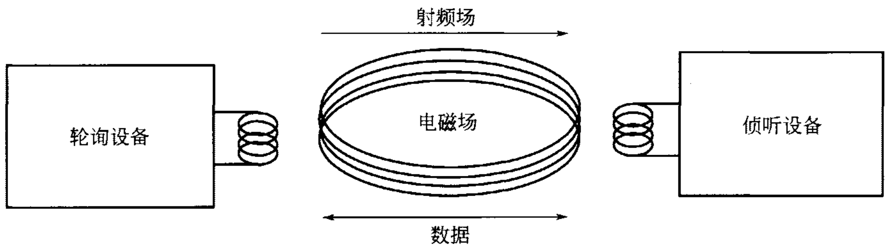


图 3-1 NFC Forum 轮询和侦听设备

成。Analog 规范中使用了振幅调制和相位调制方式（目前，相位调制并未使用）。

射频能量由轮询设备传送给侦听设备，将侦听设备激活或唤醒。数据通过调制载波的方式传递给侦听设备；侦听设备对数据进行解码处理，并通过负载调制的方式返回给侦听设备。负载调制在第 1 章介绍过，这里不再赘述。

3.1.2 NFC Forum 参考设备

本章将 NFC Forum 参考设备简称为 NFC 参考设备，原因同 NFC Forum 设备。NFC 参考设备分为参考轮询设备和参考侦听设备。NFC 参考设备可以产生各种参数以及相应的参数取值，例如某些极限条件下的参数取值，从而特性化 NFC 轮询和侦听设备。NFC 参考轮询设备能够特性化出 NFC 侦听设备，而 NFC 参考侦听设备可以特性化出 NFC 轮询设备。NFC Forum 给出了 NFC 参考设备的实现，因此 NFC 厂商可以自行设计或通过购买的方式获取 NFC 参考设备。NFC 参考设备是 NFC Forum 认为完全符合 Analog 标准的参考设备，用于评估其他 NFC 设备的射频特性参数。

NFC 参考轮询设备连接到一个合适的信号产生器和功率放大器后，就可以发送命令给侦听设备。侦听设备回复的响应可以通过测量装置捕获并进行分析。

NFC 参考侦听设备分析轮询设备发出的射频信号，如分析频率、波形等；或使用合适的信号产生器产生各种负载调制的信号，向轮询设备发送响应消息。在 NFC 参考侦听设备中，可以配置一些固定的电阻负载来模拟一些常用的和最差情况下的场景来测试轮询设备。

对于 NFC 设备的发送需求和特性来说，都可以通过 NFC 参考设备的接收天线捕捉到。也就是说，对于轮询设备的发送特性，可以通过参考侦听设备获得，而侦听设备的发送特性由参考轮询设备获得。举例来说，一个轮询设备必须提供一定的 RF 场强度（功率）给侦听设备，该射频场强度被 NFC 参考侦听设备获取并通过 J1 点的输出电压来表示。J1 点的值应当落在最大和最小接收功率范围内。

NFC 设备的接收特性是否满足需求，主要是由 NFC 参考设备对信号参数进行不同的取值，然后发送给 NFC 设备来实现的。NFC 轮询设备的接收特性是通过参考侦听设备发送不同等级或功率的负载调制信号来测量的。侦听设备的接收特性是由参考轮询设备发送不同等级或功率的调制信号来测量的。举例来说，NFC 侦听设备必须处于轮询设备产生的一定功率的射频场中，NFC 参考轮询设备将产生不同功率的射频场，其强度在最大和最小阈值之间变化。

需要注意的是，NFC 参考轮询设备产生的信号功率实际上根据参考侦听设备而确立，也就是说，这个值是由参考侦听设备测量得到的，并且反馈给参考轮询设备。参考轮询设备通过增加或减少功率的方式来确保参考侦听设备能够接收到正确的功率值。

图 3-2 描述了 NFC 参考轮询设备和 NFC 参考侦听设备。从图中可以看出，对于 NFC 参考轮询设备来说，有两个参考点 J1 和 J2。J1 点是 NFC 轮询设备产生的射频场和振幅调制的信号

参考点，J2 点为 NFC 参考轮询设备从 NFC 侦听设备处获取的负载调制信号的参考点。

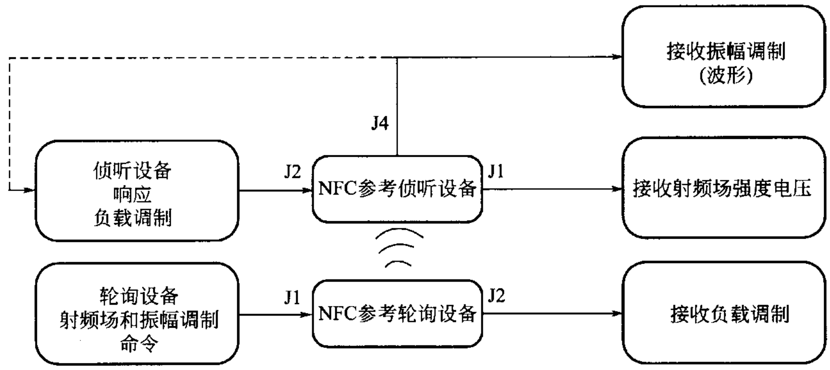


图 3-2　NFC 参考轮询设备和 NFC 参考侦听设备

NFC 参考侦听设备有三个参考点，即 J1、J2 和 J4。J1 点为 NFC 参考侦听设备从轮询设备处获得的射频场强度的参考点，J4 点为 NFC 参考侦听设备从轮询设备处获取的振幅调制信号的参考点，J2 点为参考侦听设备响应轮询设备而产生的负载调制信号的参考点。

总而言之，NFC 参考设备与 NFC 设备之间的关系为：

① 轮询设备产生的射频场功率是由参考侦听设备观察到的。

② 轮询设备的数据传输是由参考侦听设备观察到的。

③ 轮询设备的数据接收（负载调制的敏感度）是由参考侦听设备产生多个不同的信号进行观察的。

④ 侦听设备的数据发送是由参考轮询设备观察的。

⑤ 侦听设备的数据接收和功率接收是通过参考轮询设备发送不同功率电平和调制特性的信号进行观察的，例如设置一些极限值等。

3.1.3 工作空间

工作空间是轮询设备产生的一个三维射频场空间。在此空间中，可以保证 NFC 设备间的正常交互，如图 3-3 所示。工作空间通过参考点并沿着垂直轴线来测量。这里假定轮询设

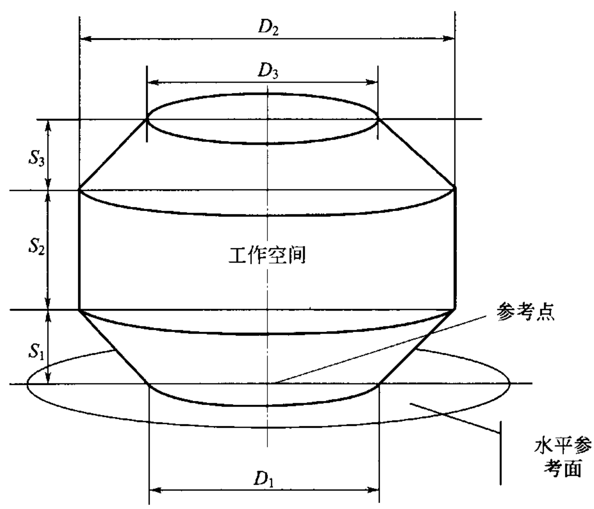


图 3-3　工作空间

备是固定不动的，而侦听设备在工作空间中是缓慢移动的。侦听设备在轮询设备工作空间中的位置是通过四元组（r，φ，z，θ）确定的。以轮询设备的位置为参考点，（r，φ，z）=（0，0，0）。

3.2 射频功率与信号接口

本节主要介绍从轮询设备到侦听设备的功率传输的参数需求。

3.2.1 功率传输需求

对于轮询设备的功率发送需求来说，轮询设备提供没有经过任何调制的射频场工作域。当参考侦听设备处于工作域中时，参考侦听设备在 J1 点产生一个输出电压 V_{ov}。将非常短时间内（>10μs）的平均值作为 V_{ov} 的定值。该值应当在 Analog 规范定义的 V_{ov} 的最大和最小范围内。

对于侦听设备功率接收的需求来说，需要参考侦听设备的配合。

① 参考轮询设备提供没有经过任何调制的射频场工作域。

② 为参考侦听设备配置一个固定的负载阻抗，并且在 J2 点上不发送负载调制信号（J2 点如图 3-2 所示）。

③ 将参考侦听设备放置在工作域中，工作域中的射频功率将传送给参考侦听设备，同时在 J1 点测得输出电压 $V_{s,ov}$。

④ 参考轮询设备进行载波调制，在参考侦听设备的 J4 点获取振幅调制信号。

⑤ 取走参考侦听设备，将侦听设备放入该工作域，确定侦听设备在 NFC-A/B/F 三种技术下是否能够获取功率正常的工作，即是否能够响应轮询设备发送的命令。

3.2.2 侦听设备对工作域的影响

侦听设备进入轮询设备的工作域后，由于引入了额外的负载，会对工作域带来一定的变化，将减少工作域中的磁场强度，因此需要对侦听设备所能提供的最大负载有一定的限制。测试步骤如下所述：

① 参考轮询设备提供没有经过任何调制的射频场工作域。

② 将侦听设备放入轮询设备提供的工作域。

③ 在参考轮询设备的 J2 点测试电压，并比较与未加入侦听设备 J2 点电压的变化值 ΔV_{ov}，该值应当在 Analog 规范中定义的最大和最小限度范围内。

3.2.3 载波频率需求

对于轮询设备的载波频率需求来说，需要将其限定在一定的范围内，如 Analog 规范给出的参考范围为 13.553～13.567MHz。具体的测试步骤如下所述：

① 将参考侦听设备配置成固定的阻抗负载，并在 J2 点不产生负载调制信号。

② 轮询设备产生载波。

③ 将参考侦听设备放入轮询设备的工作域，在参考侦听设备的 J4 点获取振幅调制信号，并恢复出载波的频率。该频率应当在 Analog 规范定义的最大和最小范围内。

对于侦听设备的载波频率需求来说，在最大和最小频率范围内，要求侦听设备功能正常，即能够响应轮询设备发送的命令。其测试方法如下所述：

① 参考轮询设备产生规定频率的载波信号。

② 参考轮询设备发送轮询命令（如 NFC-A 的 SENS _ REQ，NFC-B 的 SENSB _ REQ，NFC-F 的 SENSF _ REQ）。

③ 将侦听设备放入参考轮询设备的工作域，测试侦听设备是否能够正确地回复响应。

3.2.4 侦听设备的唤醒

在侦听设备响应轮询设备发送的轮询命令之前，轮询设备要通过射频场能量将侦听设备唤醒。侦听设备唤醒的测试方法如下所述：

① 参考轮询设备产生未经调制的射频场工作域。

② 参考轮询设备将工作域的功率设置在最大和最小限度范围内。

③ 参考轮询设备发送轮询命令。

④ 将侦听设备放入工作域。

⑤ 检测侦听设备的状态是否从 NO _ REMOTE _ FIELD 状态转变为 IDLE 状态。其中，NO _ REMOTE _ FIELD 是侦听设备未发现周围射频场时所处的状态，IDLE 为发现射频场后进入的状态。

3.2.5 轮询设备射频冲突避免

当 NFC 设备检测到周围存在射频场后，为了避免射频冲突，NFC 设备不能产生载波。对于轮询设备来说，该需求的测试方法如下所述：

① 参考轮询设备产生载波信号。

② 将参考侦听设备设置为固定的阻抗负载，并放入参考轮询设备的工作域，在参考侦听设备的 J1 点可以测得输出电压 V_s

③ 移除参考侦听设备，放入轮询设备。该轮询设备应当不产生工作域。

3.3 轮询设备到侦听设备的信号接口

本节主要介绍轮询设备到侦听设备的信号接口，即轮询设备和侦听设备之间数据通信的接口，涉及到采用 NFC-A、NFC-B 和 NFC-F 技术时轮询设备到侦听设备所采用的调制技术要求。A/B/F 三种技术均采用 ASK（Amplitude Shift Keying）调制技术，区别点主要在于采用不同的调制深度。

3.3.1 NFC-A 技术的调制需求

在 NFC-A 技术中，从轮询设备到侦听设备的调制方式采用 100％的 ASK 调制。载波被打开或关闭，关闭时表示低电平，打开时表示高电平。但是在实际应用中，可能存在 95％或更高的调制深度，测试步骤如下所述：

① 参考侦听设备被设置为固定的阻抗负载，并且 J2 点不产生负载调制信号。

② 轮询设备发送轮询命令 ALL _ REQ 或 SENS _ REQ。

③ 将参考侦听设备放入轮询设备的工作域。

④ 参考侦听设备接收到轮询设备发送的 ALL _ REQ 或 SENS _ REQ 消息后，从 J4 点分析特性化。

如图 3-4 所示，从 J4 点输出的波形必须符合以下特征。

V 代表从 J4 点监视到的信号的包络，V_1 表示调制前测量到的初始值，$V_2=0.05V_1$，

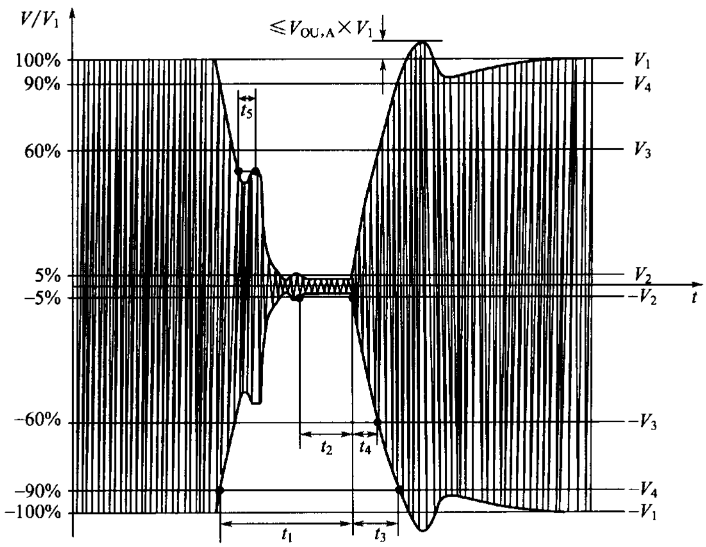


图 3-4 NFC-A 技术轮询设备到侦听设备的调制需求

$V_3=0.6V_1$，$V_4=0.9V_1$。包络的下降沿是从 V_4 降到了 V_2，包络的上升沿是 V_2 上升到了 V_4。具体要求如下所述：

① 在下降沿的 V_4 到上升沿之前 V_2 之间的时间为 t_1。

② 如果包络 V 没有从 V_4 到 V_2 单调下降，那么包络上升到最大值与之前下降时的相同值之间的时间为 t_5。

③ 下降沿之后的上升沿应当保持在 $V_{OU,A}\times V_1$ 之内。

④ 在 t_2 时间内，V 应当保持在 V_2 以下。

⑤ 在 V_2 到 V_3 的时间 t_4 内，V 应当单调上升。

⑥ 在 V_2 到 V_4 的时间 t_3 内，V 应当单调上升。

⑦ 上升沿过程中的突发脉冲不能超过 $(1\pm V_{OU,A})\times V_1$。

关于 t_1、t_2、t_3、t_4、t_5 和 $V_{OU,A}$ 的取值范围，Analog 规范有详尽定义，可以参考其附录。

对于侦听设备来说，从轮询设备到侦听设备的调制应当保证侦听设备能够正常工作，即能够正确地响应轮询设备发送的命令。

3.3.2 NFC-B 技术的调制要求

在 NFC-B 技术中，从轮询设备到侦听设备的调制方式采用 10% 的 ASK 调制技术，通过调制指数来控制载波的振幅。测试步骤如下所述：

① 参考侦听设备被设置为固定的阻抗负载，并且 J2 点不产生负载调制信号。

② 轮询设备发送轮询命令 ALLB _ REQ 或 SENSB _ REQ。

③ 将参考侦听设备放入轮询设备的工作域。

④ 参考侦听设备接收到轮询设备发送的 ALLB _ REQ 或 SENSB _ REQ 消息后，从 J4 点分析特性化。

如图 3-5 所示，从 J4 点输出的波形必须符合以下特征。

V 代表从 J4 点监视到的信号的包络，V_a 表示调制前测量到的初始值，V_b 表示低电平，调制指数 m_i、V_3 和 V_4 分别定义为

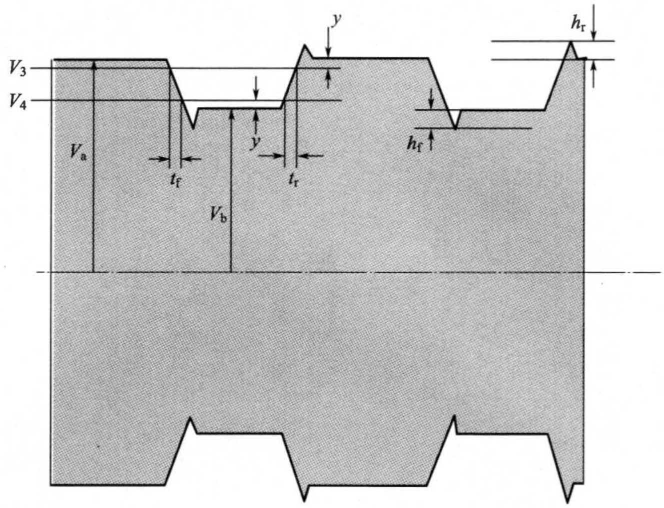


图 3-5 NFC-B 技术轮询设备到侦听设备的调制需求

$$m_i = \frac{V_a - V_b}{V_a + V_b}$$

$$V_3 = V_a - 0.1(V_a - V_b)$$

$$V_4 = V_b + 0.1(V_a - V_b)$$

NFC-B 技术下轮询设备到侦听设备的调制需求如下所述：

① 调制指数 m_i 应当为 $mod_{i,B}$。

② 包络从 V_3 到 V_4 下降的时间为 $t_{f,B}$。

③ 包络从 V_4 到 V_3 上升的时间为 $t_{r,B}$。

④ 上升沿和下降沿应当是单调变化的。

⑤ 下降沿 h_f 的过程中，正、负突发脉冲应当小于 $V_{OU,B} \times (V_a - V_b)$。

⑥ 上升沿 h_r 的过程中，正、负突发脉冲应当小于 $V_{OU,B} \times (V_a - V_b)$。

关于 $mod_{i,B}$、$t_{f,B}$、$t_{r,B}$和 $V_{OU,B}$的取值范围，Analog 规范做了详尽定义，可以参考其附录。

对于侦听设备来说，从轮询设备到侦听设备的调制应当保证侦听设备能够正常工作，即能够正确地响应轮询设备发送的命令。

3.3.3 NFC-F 技术的调制要求

在 NFC-F 技术中，从轮询设备到侦听设备的调制方式采用 10% 的 ASK 调制技术，通过调制指数来控制载波的振幅。测试步骤如下所述：

① 参考侦听设备被设置为固定的阻抗负载，并且 J2 点不产生负载调制信号。

② 轮询设备发送轮询命令 SENSF _ REQ。

③ 将参考侦听设备放入轮询设备的工作域。

④ 参考侦听设备接收到轮询设备发送的 SENSF _ REQ 消息后，从 J4 点分析特性化。

如图 3-6 所示，从 J4 点输出的波形必须符合以下特征。

V 代表从 J4 点监视到的信号的包络，V_a 表示调制前测量到的初始值，V_b 表示低电平，调制指数 m_i、V_3 和 V_4 分别定义为

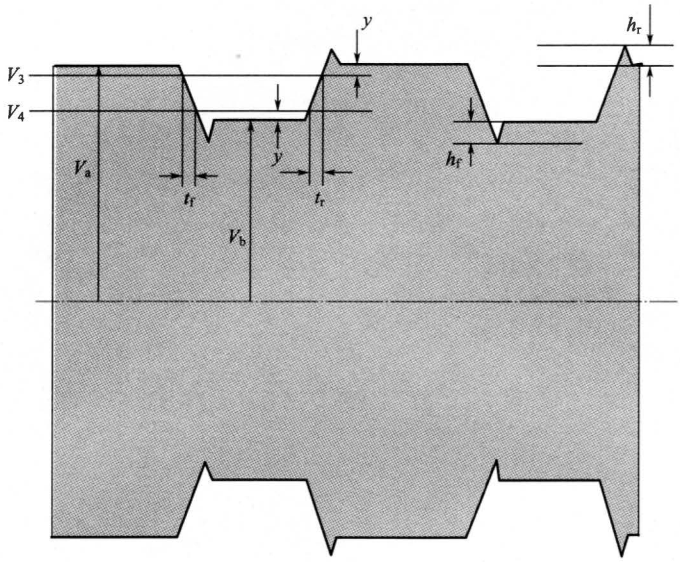


图 3-6 NFC-F 技术轮询设备到侦听设备的调制需求

$$m_i=\frac{V_a-V_b}{V_a+V_b}$$

$$V_3=V_a-0.1(V_a-V_b)$$

$$V_4=V_b+0.1(V_a-V_b)$$

NFC-F 技术下轮询设备到侦听设备的调制需求如下所述：

① 调制指数 m_i 应当为 $mod_{i,F}$。

② 包络从 V_3 到 V_4 下降的时间为 $t_{f,F}$。

③ 包络从 V_4 到 V_3 上升的时间为 $t_{r,F}$。

④ 上升沿和下降沿应当是单调变化的。

⑤ 下降沿 h_f 的过程中，正、负突发脉冲应当小于 $V_{OU,F}\times(V_a-V_b)$。

⑥ 上升沿 h_r 的过程中，正、负突发脉冲应当小于 $V_{OU,F}\times(V_a-V_b)$。

关于 $mod_{i,F}$、$t_{f,F}$、$t_{r,F}$和 $V_{OU,F}$的取值范围，Analog 规范做了详尽定义，可以参考其附录。

对于侦听设备来说，从轮询设备到侦听设备的调制应当保证侦听设备能够正常工作，即能够正确地响应轮询设备发送的命令。

3.4 侦听到轮询的信号接口

3.4.1 侦听设备负载调制的通用要求

由于主动通信模式还未正式加入 Analog 规范，因此侦听设备向轮询设备发送数据是通过负载调制的方式。对于 NFC-A 和 NFC-B，载波频率 f_c 用于产生副载波 $f_s=f_c/16$（约为 847kHz）。对于 NFC-F，采用的是 50%频宽比的负载调制。侦听设备的负载调制如图3-7所示。NFC-A 通过 OOK（On-Off Keying）的方式调制副载波；NFC-B 采用 BPSK（Binary Phase Shift Keying）的方式调制副载波；NFC-F 的负载调制方式与轮询设备到侦听

设备的调制方式一致，都采用10%ASK。通用的测试方法如下所述：

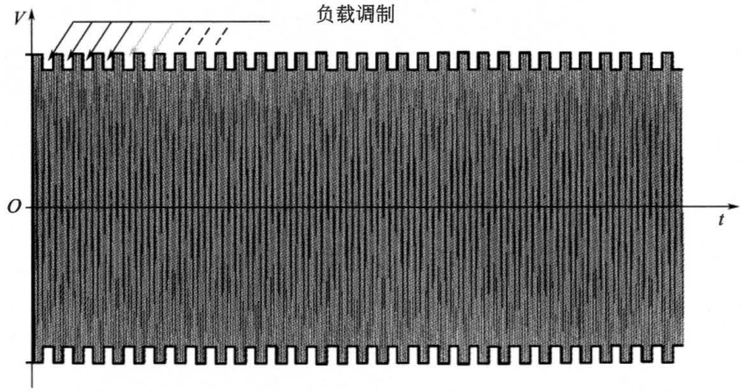


图 3-7 侦听设备到轮询设备的负载调制

① 参考轮询设备根据不同的技术（NFC-A/B/F）发送不同的轮询命令。

② 将侦听设备放入参考轮询设备的工作域。

在参考轮询设备的J2点应当能够观测到以下特性：

① 对于NFC-A和NFC-B来说，副载波频率应当为$f_c/16$。

② 对于NFC-F来说，调制信号在同步时的频率应当为$f_c/32$或$f_c/64$，在其他时间为$f_c/32$、$f_c/64$或$f_c/128$。

③ 参考轮询设备J2点调制信号的幅度V_{pp}应当为峰值V_{pp}。

关于V_{pp}的取值范围，Analog规范做了详尽定义，可以参考其附录。

3.4.2 NFC-A副载波负载调制要求

对于NFC-A技术侦听设备的负载调制方式，有以下特性要求：

① 侦听设备应当使用OOK进行副载波调制。

② 在调制副载波时，侦听设备只能在副载波的上升沿或下降沿开始比特周期。这样，调制信号可以与副载波有一个固定的相位关系，如图3-8所示。

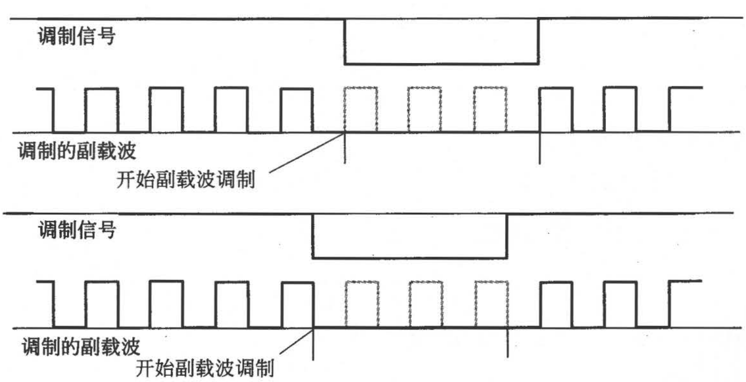


图 3-8 NFC-A 副载波调制

3.4.3 NFC-B副载波负载调制要求

NFC-B中使用BPSK进行副载波调制。在侦听设备向轮询设备发送信息之前，侦听设

备和轮询设备首先建立一个参考相位，然后侦听设备才能开始副载波调制。NFC-B 侦听设备的副载波调制有以下要求：

① NFC-B 的侦听设备通过 BPSK 进行副载波调制。

② 侦听设备只有在需要发送数据时才会产生副载波。

③ 相移只能发生在副载波的上升沿或下降沿，如图 3-9 所示。

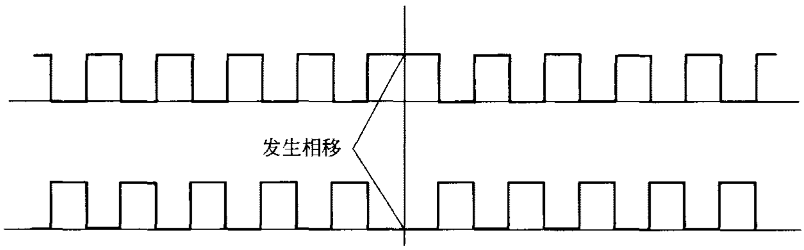


图 3-9 NFC-B 副载波调制

3.4.4 NFC-F 副载波负载调制要求

对于 NFC-F 技术侦听设备的负载调制，要求侦听设备采用 ASK 的负载调制方式。

3.5 小结

Analog 规范定义了 NFC 设备射频的模拟特性。它没有给出天线的设计，主要是通过一个外部的观测设备来评估 NFC 设备射频模拟特性参数，指导天线设计人员参数取值。本章首先解释了 Analog 规范中定义的 NFC Forum 设备和 NFC Forum 参考设备，然后介绍射频功率接口中的参数需求，最后阐述轮询设备到侦听设备的信号接口以及侦听设备到轮询设备的信号接口。本章所列参数在 Analog 规范中都有具体的取值要求，可以参考 NFC Forum-TS-Analog-1.0。

第4章 Digital/Activity规范

4.1 Digital/Activity 规范概述

4.1.1 Digital 规范

Digital 规范[1]规定了 NFC-A、NFC-B 和 NFC-F 三种技术的编码格式、调制方式、传输速率、帧格式、传输协议和命令集。同时，Digital 规范覆盖了 NFC 设备的四种角色：发起方、目标方、读写器和卡模拟设备。

发起方（Initiator）和目标方（Target）是 ISO 18092 中的称谓。本章在介绍 NFC-DEP 的小节中将沿用该称谓；而在其他小节中，为了简便起见，将 NFC 设备或标签统称为 NFC 设备。NFC 设备分为轮询设备和侦听设备。轮询设备是工作在轮询模式下的 NFC 设备，如读写器；侦听设备是工作在侦听模式下的 NFC 设备，如 NFC 标签或卡模拟设备。

本章主要针对 Digital 1.0 来介绍，因此未涉及与主动通信相关的内容。

4.1.2 Activity 规范

Digital 规范定义了通信所需要的各种参数和条件，但是没有描述如何在两个 NFC 设备间（本章中将“设备间”以及“设备与标签间”统称为“设备间”）建立通信的流程。Activity 规范[2]定义了各种“活动（Activities）”，用于描述如何通过 Digital 中定义的帧格式、命令等来建立通信。

4.1.3 Digital/Activity 架构

4.1.3.1 Digital 与 Activity 的关联

Activity 规范实际上可以认为是 Digital 规范的一个补充协议，只有将 Activity 和 Digital 放在一起来介绍，才能使读者清楚地了解 NFC 通信建立的流程。因此本章将 Digital 规范和 Activity 规范穿插在一起介绍。在介绍完 Digital 规范中的帧格式、命令集后，通过讲解 Activity 规范，来让读者了解这些命令集是如何使用的，即如何通过这些命令来完成通信。

Activity 规范中定义了 NFC 通信的四个阶段：初始化、设备激活、数据交换和设备去激活。其中，初始化包括射频冲突避免、技术发现、冲突检测等过程。在 Digital 规范中定义了这四个阶段所需的通信机制，如编码格式、调制方式、传输速率、帧格式、传输协议和

---

[1] NFC Forum. NFC Digital Protocol 1.0.

[2] NFC Forum. NFC Activity Specification 1.0.

命令集等。图 4-1 描述了 Digital 规范与 Activity 规范定义内容的关联性。

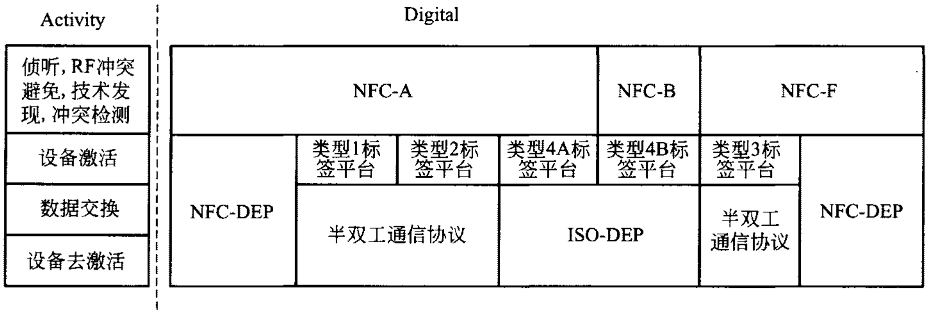


图 4-1 Digital 规范与 Activity 规范

4.1.3.2 Digital 规范数据封装

Digital 规范中定义的三种技术有着不同的调制方式、编码格式、传输速率、帧格式和命令集。本小节先介绍 NFC 通用的数据封装格式，然后总结了三种技术的调制方式、编码格式及传输速率。对于三种技术具体的帧格式、命令集，将分别详细介绍。

NFC 设备间交互的信息称为数据包。数据包一般以 SOD（Start of Data）开始，以 EOD（End of Data）结束。数据包被封装在帧中，帧以 SOF（Start of Frame）开始，以 EOF（End of Frame）结束。帧被封装在序列（Sequence）中，序列以 SOS（Start of Sequence）开始，以 EOS（End of Sequence）结束，如图 4-2 所示。

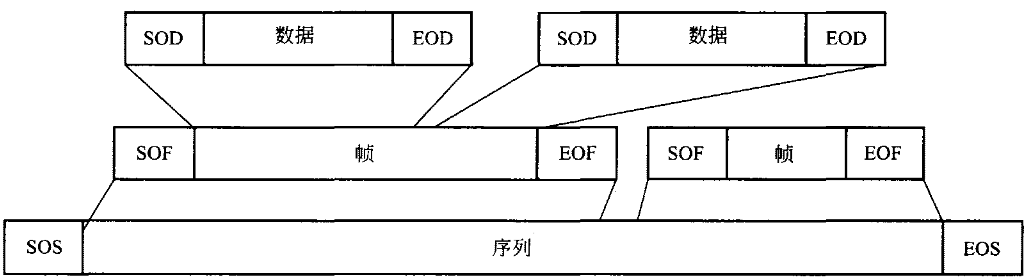


图 4-2 NFC 数据封装格式

SOD、EOD、SOF 和 EOF 是可选的，例如 NFC-A 中就不需要 SOD。序列中的 SOF 和 EOF 是射频信号的模式，用于射频信号发送和接收的同步。举例来说，接收设备收到的射频信号中如果包含 SOF 的信号波形，就认为这是一个序列的开始。三种技术对于帧的格式有不同的分类和定义，下面将分别介绍。

4.1.3.3 NFC 调制及编码方式

三种 NFC 技术的调制方式和编码方式如表 4-1 所示。

表 4-1 三种 NFC 技术的调制方式与编码方式对比

技　术	轮询设备到侦听设备		侦听设备到轮询设备	
	调制方式	编码方式	调制方式	编码方式
NFC-A	100% ASK	改进型 Miller 编码	OOK 副载波调制	曼彻斯特编码
NFC-B	10% ASK	NRZ-L 编码	BPSK	NRZ-L 编码
NFC-F	10% ASK	曼彻斯特编码	10% ASK	曼彻斯特编码

4.1.3.4 NFC 通信流程

Activity 规范定义了 NFC 通信的四个阶段，分述如下。

(1) 初始化阶段

Digital 规范定义了三种技术 NFC-A、NFC-B 和 NFC-F。这三种技术都定义了各自的初始化过程，也定义了 NFC 通信的基本数据格式、传输速率、编码方式和调制方式。这三种技术是 NFC 通信的基础，后面提到的 NFC-DEP、类型 1～4 标签平台、ISO-DEP 等，都以这三种技术为基础。初始化过程中的射频冲突避免和技术发现与三种技术无关，是 NFC 设备的通用过程。而对于冲突检测过程来说，每一个技术都有不同的实现方式，详细内容将在后面详细介绍。

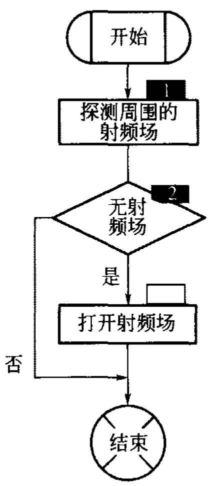


图 4-3 射频冲突避免

射频冲突避免是指在 NFC 设备产生射频场前，需要检测周围的射频场。如果周围存在射频场，NFC 设备将不会产生射频场，如图 4-3 所示。

技术发现是指 NFC 轮询设备发送某种技术的轮询命令，来发现周围支持该技术的侦听设备，具体流程如图 4-4 所示。当 NFC 轮询发送某种技术的轮询命令后，只有支持该技术的侦听设备才会响应，不支持该技术的侦听设备不会响应。NFC 轮询设备根据配置的参数情况，完成技术发现的过程。图 4-4 中的判决点“是否进行 NFC-X 技术发现”和“是否继续进行发现”都是通过参数判断的。该参数一般由 NFC 芯片厂商在出厂时预设，开发人员在使用过程中可以修改，以便控制 NFC 芯片只针对某种技术进行发现。一般情况下，技术发现是按照 A、B、F 的顺序来进行。如果 NFC 芯片支持私有技术，也可以对该私有技术进行发现。

技术发现的过程只是发现支持该技术的侦听设备是否存在，不会将这些侦听设备识别出来。对于侦听设备的识别，在冲突检测流程中介绍。

(2) 设备激活

设备激活是指 NFC 设备之间进行参数协商，协商成功后可以进行数据通信。对于采用 NFC-DEP 作为传输协议的 NFC 设备，设备激活的命令包含在 NFC-DEP 中。为了支持读写模式和卡模拟模式，Digital 规范中定义了四种类型的标签平台。标签平台属于技术的子集，如类型 1、类型 2 和类型 4A 属于 NFC-A 的子集，类型 4B 属于 NFC-B 的子集，类型 3 属于 NFC-F 的子集。在设备激活的过程中，四类标签平台有其特有的激活命令（实际上，对于类型 1/2/3 标签平台，没有设备激活命令）。

(3) 数据传输

在设备激活之后，NFC 设备之间就可以传输数据了。支持点对点通信的设备采用 NFC-DEP 作为传输协议来交换数据，支持 ISO 14443 的读写器或卡模拟采用 ISO-DEP 作为传输协议来传输数据；对于支持类型 1/2/3 标签平台的读写器或卡模拟，采用半双工通信方式交换数据。

(4) 设备去激活

在数据通信完毕后，进行设备的去激活，从而结束通信。

4.1.4 Digital/Activity 规范与 ISO 标准

第 2 章大致介绍了 NFC Forum 规范与 ISO 标准的区别与联系，但是没有详细地比较。在了解 NFC Forum Digital 及 Activity 规范之后，读者应对 NFC Forum 与 ISO 之间的标准

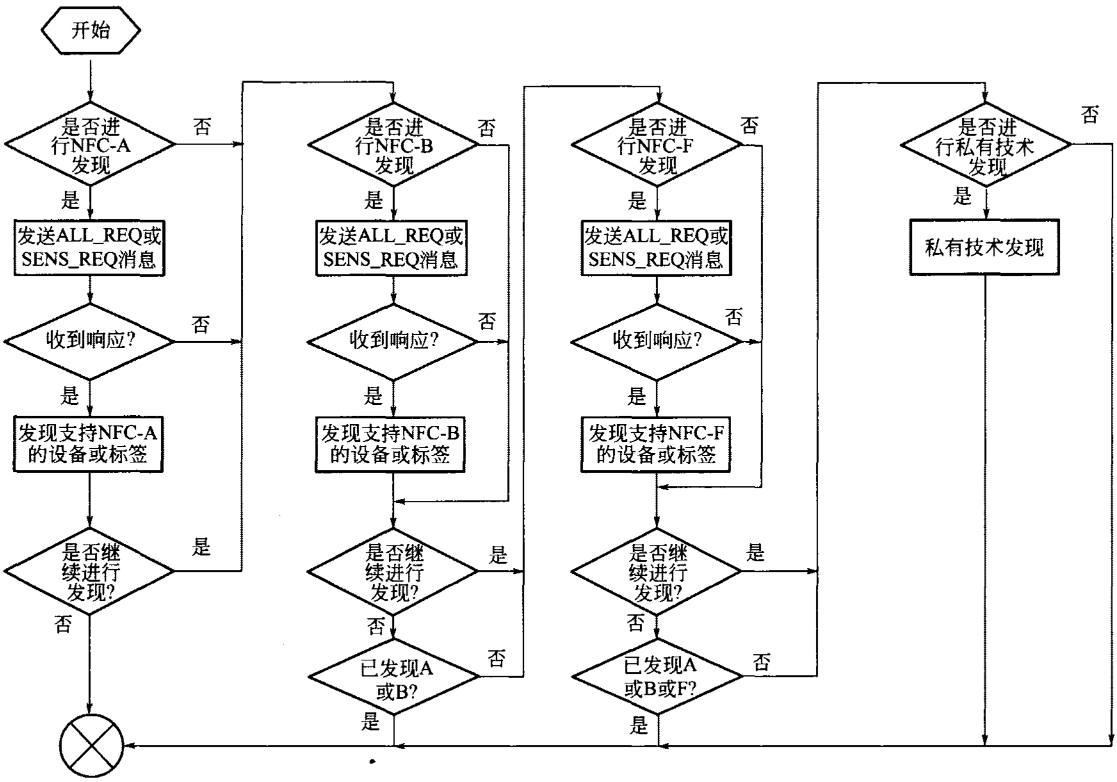


图 4-4　技术发现过程

规范的区别和联系有更深入的理解，如图 4-5 所示。

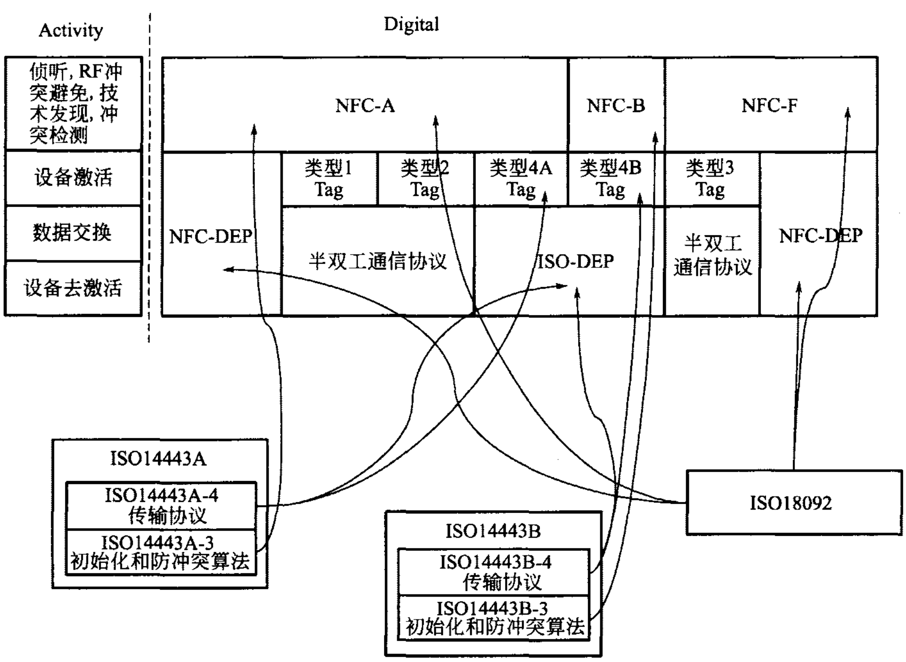


图 4-5　NFC Forum Digital 规范与 ISO 标准

4.2 NFC-A

4.2.1 帧格式

NFC-A 技术定义了三种类型的帧：短帧（Short Frame）、标准帧（Standard Frame）和面向比特的 SDD 帧（Bit-oriented Single Device Detection Frame）。

短帧用于通信的初始化，其帧格式如图 4-6 所示。从轮询到侦听的通信，SOF 和 EOF 均设置为逻辑“0”；从侦听到轮询的通信，SOF 设置为逻辑“1”；EOF 设置为模式 F（即 NFC-A 的信号模式 F，具体可参见 Digital 规范）。

	lsb						msb	
SOF	b1	b2	b3	b4	b5	b6	b7	EOF

图 4-6 短帧的帧格式

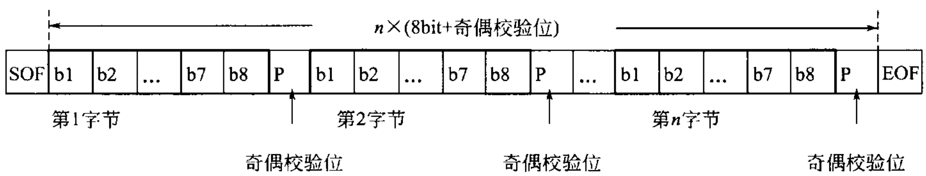


图 4-7 标准帧的帧格式

标准帧用于数据交换，其帧格式如图 4-7 所示。在标准帧中，每 8 个比特后跟一个奇偶校验位。包含在标准帧中的数据格式如图 4-8 所示，不含 SOD，EOD 为 2 字节（Byte）的 NFC-A 循环校验码。需要注意的是，包含在短帧和 SDD 帧中的数据没有 SOD 和 EOD。

数据(命令或响应)				[EOD]	
Byte 1	Byte 2	...	Byte *n*	[CRC_A1]	[CRC_A2]

图 4-8 NFC-A 数据格式

面向比特的 SDD 帧，本书中简称为 SDD 帧，用于冲突检测过程。SDD 帧实际上是把一个 7 字节长的标准帧分成了两部分，分割点可以发生在任何比特位上。SDD 帧的每一部分可能不是字节的整数倍，因此称为面向比特的帧。关于 SDD 帧的用法，如何分割，将在冲突检测流程中详细介绍。图 4-9 描述了在第 2 字节的第 1 个比特（bit，位）之后分割的例子。

第一部分的 SDD 帧是由轮询设备发送的；第二部分的 SDD 帧由侦听设备发送，用来响应第一部分的 SDD 帧。第一部分和第二部分的 SDD 帧加起来一共 56bit。第一部分 SDD 帧最小长度为 16bit，最大长度为 55bit，因此相对应的第二部分 SDD 帧的最小长度为 1bit，最大长度为 40bit。如果分割点刚好处于一个字节的第 8 个 bit（比特）之后，相应的奇偶校验位应当加入到第一部分最后一个 bit 后。如果分割点在一个字节的中间，不需要在第一部分的最后一个 bit 后加奇偶校验位，该奇偶校验位将出现在第二部分中。

4.2.2 命令

NFC-A 的命令集如表 4-2 所示。

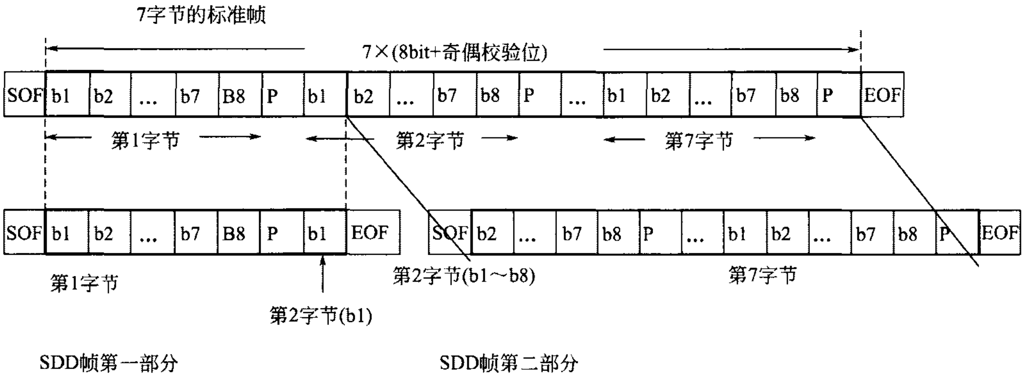


图 4-9 SDD 帧的帧结构

表 4-2 NFC-A 命令集

命令	响应	包含 EOD	帧类型
ALL_REQ,SENS_REQ		否	短帧
	SENS_RES	否	标准帧
SDD_REQ		否	SDD 帧
	SDD_RES	否	SDD 帧
SEL_REQ		是	标准帧
	SEL_RES	是	标准帧
SLP_REQ		是	标准帧

4.2.2.1 ALL_ REQ 和 SENS_ REQ

ALL _ REQ 和 SENS _ REQ 命令用于技术发现过程，由轮询设备发送，用于探测支持 NFC-A 技术的侦听设备，如图 4-4 所示。ALL _ REQ 和 SENS _ REQ 的主要区别是：ALL _ REQ用于唤醒已经被检测出，但是处于睡眠的侦听设备；SENS _ REQ 用于探测还未被检测出的侦听设备。

ALL _ REQ 的消息格式如图 4-10 所示。

b7	b6	b5	b4	b3	b2	b1	描述
1	0	1	0	0	1	0	52h=ALL_REQ

图 4-10 ALL _ REQ 消息格式

SENS _ REQ 的消息格式如图 4-11 所示。

b7	b6	b5	b4	b3	b2	b1	描述
0	1	0	0	1	1	0	26h=SENS_REQ

图 4-11 SENS _ REQ 消息格式

4.2.2.2 SENS_ RES

SENS _ RES 用于响应 ALL _ REQ 或 SENS _ REQ 命令，由侦听设备发送，属于标准帧，由两个字节组成，其第一个字节的格式如图 4-12 所示。

其中，b8 和 b7 用于标识 NFC-A 技术中 3 种不同长度的 NFCID（NFC 标示符），即 4

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0							NFCID1 长度为4字节
0	1							NFCID1 长度为7字节
1	0							NFCID1 长度为10字节
1	1							预留
		0						预留
			1	0	0	0	0	Bit frame SDD
			0	1	0	0	0	Bit frame SDD
			0	0	1	0	0	Bit frame SDD
			0	0	0	1	0	Bit frame SDD
			0	0	0	0	1	Bit frame SDD
			0	0	0	0	0	Bit Frame SDD(用于鉴别类型1标签平台)
			其他值					预留

图 4-12 SENS _ RES 第一个字节的格式

字节、7 字节和 10 字节。NFCID 用于唯一地标识一个 NFC 设备。NFC-A 技术中的 NFCID 称为 NFCID1，NFC-B 中称为 NFCID0，NFC-F 中称为 NFCID2。需要注意的是，在 NFC-A 技术中，NFCID1 只用于在冲突检测过程中唯一地标识一个 NFC 设备。

b5～b1 用于描述侦听设备是否支持冲突检测。如果侦听设备发送的 SENS _ RES 消息中 b5～b1 的值如图 4-12 所示，那么该侦听设备支持冲突检测。这里提及的支持冲突检测，是指侦听设备能够响应轮询设备在冲突检测过程中发送的 SDD _ REQ 消息。另外，如果 b5～b1 取值为全零，表明该侦听设备支持类型 1 标签平台，即该设备可以模拟成类型 1 的标签，这个取值的目的是为了让轮询设备能够快速地识别出支持类型 1 的标签平台。因为对于 NFC-A 技术来说，响应轮询设备发送的 SENS _ REQ 消息的侦听设备可以是类型 1、类型 2、类型 4A 的标签，或支持 NFC-DEP 协议的设备。

第二个字节的格式如图 4-13 所示。第二个字节的 b4～b1 如果设置为 1100，表示该侦听设备模拟成类型 1 的标签。如果第一个字节中 b5～b1 设置成 00000，那么第二个字节的 b4～b1 必须设置为 1100，否则轮询设备会认为该 SENS _ RES 响应消息是错误的；反之，如果第二个字节的 b4～b1 设置为 1100，那么第一个字节的 b5～b1 必须是 00000。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0	0	0					预留
				1	1	0	0	类型1标签平台配置
				其他值				其他配置

图 4-13 SENS _ RES 第二个字节的格式

4.2.2.3 SDD_ REQ

SDD _ REQ 命令由轮询设备发送，用于冲突检测过程。在 NFC-A 技术中，冲突检测又称为单设备检测（Single Device Detection）。冲突检测是指轮询设备周围可能存在多个侦听设备，当轮询设备发送 SENS _ REQ 命令进行探测时，收到 SENS _ REQ 命令的支持 NFC-A 技术的侦听设备都会响应 SENS _ RES 消息。这样，轮询设备无法识别出侦听设备，需要通过冲突检测过程，即发送 SDD _ REQ 命令来识别每一个侦听设备。

SDD _ REQ 命令格式如图 4-14 所示。

Byte 1	Byte 2	n数据位(比特)
SEL_CMD	SEL_PAR	b1…bn

图 4-14　SDD _ REQ 命令格式

SEL _ CMD 用于指示 NFCID1 的级联级别（Cascade Level，CL），其数据格式如图4-15所示。SEL _ CMD 用于轮询设备指定侦听设备在 SDD _ RES 中回复的 NFCID1 的级联级别。

b8	b7	b6	b5	b4	b3	b2	b1	描述
1	0	0	1	0	0	1	1	93h：SDD_REQ CL1
1	0	0	1	0	1	0	1	95h：SDD_REQ CL2
1	0	0	1	0	1	1	1	97h：SDD_REQ CL3
其他值								预留

图 4-15　SEL _ CMD 格式

NFCID1 用于在 NFC-A 技术的冲突检测过程中唯一地标识一个侦听 NFC 设备，因此冲突检测的过程就是识别出每个侦听设备 NFCID1。NFCID1 有 3 种长度：4 字节、7 字节和 10 字节。为了统一地处理 3 种不同长度的 NFCID1，就有了级联级别，每个级联包括 4 个字节。NFCID1 被分成 3 个级联级别，即将 NFCID1 分成 3 个部分。对于 4 字节的 NFCID1，只有 1 个级联级别。对于 7 字节 NFCID1，有 2 个级联级别，即 NFCID1 被分成 2 个部分，前 3 个字节为级联级别 1，后 4 个字节为级联级别 2。对于 10 字节的 NFCID1，有 3 个级联级别，前 3 个字节为级联级别 1，中间 3 个字节为级联级别 2，后 4 个字节为级联级别 3。具体来说，对于 7 字节的 NFCID1，级联级别 1 中会加入一个填充字节，与 NFCID1 的前 3 个字节构成 4 字节。对于 10 字节的 NFCID1，在级联级别 1 和级联级别 2 中分别加入一个填充字节，凑成 4 字节。每轮冲突检测针对一个级联级别进行处理，解决了对不同字节长度 NFCID1 的统一处理。

SEL _ PAR 指示 SDD _ REQ 命令的长度，包括 SEL _ CMD 和 SEL _ PAR 的长度（共 2 字节），因此它也可以指示出 SEL _ PAR 后数据位的个数。SEL _ PAR 的 b8～b5 用于指示 SEL _ REQ 命令长度的字节数，其格式如图 4-16 所示，这里是指整字节数。

b8	b7	b6	b5	描述
0	0	1	0	2字节
0	0	1	1	3字节
0	1	0	0	4字节
0	1	0	1	5字节
0	1	1	0	6字节
0	1	1	1	7字节

图 4-16　SEL _ PAR b8～b5

b4	b3	b2	b1	描述
0	0	0	0	比特数=0
0	0	0	1	比特数=1
0	0	1	0	比特数=2
0	0	1	1	比特数=3
0	1	0	0	比特数=4
0	1	0	1	比特数=5
0	1	1	0	比特数=6
0	1	1	1	比特数=7

图 4-17　SEL _ PAR b4～b1

SEL _ PAR 的 b4～b1 用于指示 SDD _ REQ 命令中的位（比特）数，如图 4-17 所示。

前面已经提到，SDD _ REQ 命令是面向比特的 SDD 帧，由于可以在任何位置被分割，因此其长度可能不是完整的字节数，SEL _ PAR 的 b4～b1 就是用于描述多出来的位数。当 b4～b1 为 0000 时，表示 SDD _ REQ 命令拥有完整的字节长度，否则表示长度为非完整字节。

跟随在 SEL _ PAR 之后的数据位是指在 SEL _ CMD 指示的级联级别下，已经被识别出的 NFCID1 的前 n 个比特。

4.2.2.4 SDD_ RES

侦听设备收到 SDD _ REQ 命令后，将比较自己的 NFCID1 与 SDD _ REQ 中的数据位。如果在相应的级联级别下（即 SEL _ CMD 中指示的级联级别）二者相匹配，目标设备将回复 SDD _ RES 响应消息，否则目标设备将忽略 SDD _ REQ 命令。

当轮询设备第一次发送 SDD _ REQ 命令时，SEL _ CMD 指示为级联级别 1，SEL _ PAR 指示为 2 字节，比特数为 0，即 SDD _ REQ 命令中只包含 SEL _ CMD 和 SEL _ PAR 2 个字节，SEL _ PAR 后面没有任何数据位，因为冲突检测过程还未识别出任何侦听设备的 NFCID1 值。当侦听设备收到该 SDD _ REQ 命令后，发现该命令没有包含任何数据位，侦听设备将向轮询设备返回 SDD _ RES 消息。该消息中将包含级联级别 1 的 NFCID1 值，如图 4-18 所示。这里需要注意，图 4-18 所示的 SDD _ RES 消息格式是响应 SEL _ PAR 设置为 20h 的 SDD _ REQ 命令。$nfcid1_n$ 是指 NFCID1 的第 n 个字节。CT（Cascade Tag，级联标签）就是前面提到的填充字节，将一个级联级别凑成 4 字节。SDD _ RES 响应消息的最后还包含 1 字节的 BCC，该字节是前面 4 字节的异或，用于校验。因此，SDD _ RES 响应消息中一共包含 5 个字节。NFCID1 一般由设备自己动态产生，当设备从关闭（Power-off）状态到空闲（IDLE）状态时，将重新动态生成 NFCID1。对于 4 字节的 NFCID1，$nfcid1_0$ 固定为 08h，$nfcid1_1$～$nfcid1_3$ 为动态产生。对于 7 字节的 NFCID1，$nfcid1_3$ 不能被设置为 88h。另外，CT 不能被设置为 88h。

SEL_CMD	NFCID1的长度	SDD_RES
93h	4	CL1：$nfcid1_0$ $nfcid1_1$ $nfcid1_2$ $nfcid1_3$ BCC
93h	>4	CL1：CT $nfcid1_0$ $nfcid1_1$ $nfcid1_2$ BCC
95h	7	CL2：$nfcid1_3$ $nfcid1_4$ $nfcid1_5$ $nfcid1_6$ BCC
95h	>7	CL2：CT $nfcid1_3$ $nfcid1_4$ $nfcid1_5$ BCC
97h	10	CL3：$nfcid1_6$ $nfcid1_7$ $nfcid1_8$ $nfcid1_9$ BCC

图 4-18 SDD _ RES 响应消息格式

如果 SDD _ REQ 命令中的 SEL _ PAR 不为 20h，即 SEL _ PAR 字节后面包含数据位，那么 SDD _ RES 响应消息的格式如图 4-19 所示。当侦听设备收到 SDD _ REQ 命令后，根据 SEL _ CMD 指示的级联级别，判别 SEL _ PAR 后的数据位与自己 NFCID1 的前 n 个比特是否一致。如果一致，就发送 SDD _ RES 响应消息。SDD _ RES 响应中包含该级联级别剩余的数据比特。如图 4-19 所示，SEL _ CMD＝93h，指示级联级别为 1。SEL _ PAR＝36h，表示 SEL _ PAR 后面包含 14 个数据位（1 字节＋6bit）。如果侦听设备 NFCID1 的前 14bit 与 SDD _ REQ 命令中的这 14bit 相等，侦听设备将发送 SDD _ RES 来响应，响应消息中将包含 NFCID1 中剩余的 26bit。

总而言之，SDD _ REQ 和 SDD _ RES 中的数据位加起来是 40bit，即构成一个完整的级联级别 4 字节和 1 字节 BCC。这也是前面提到的面向比特的 SDD 帧，实际上是一个 7 字节的标准帧被拆分成两部分，第一部分为 SDD _ REQ，第二部分即 SDD _ RES。至于为什么

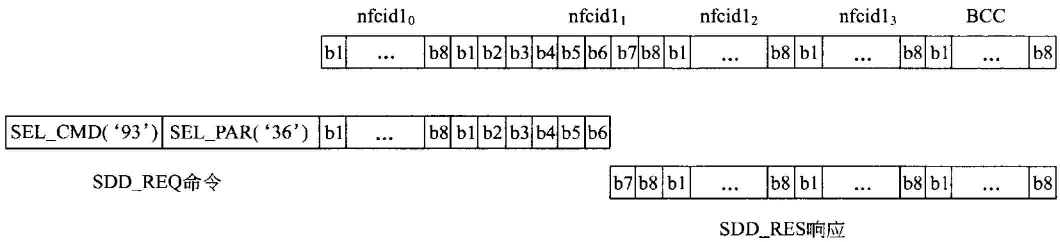


图 4-19 SDD _ RES 举例

这样设计，以及 SDD _ REQ 和 SDD _ RES 如何进行冲突检测，将在 4.2.3 节中介绍。

4.2.2.5 SEL_ REQ/SEL_ RES

SEL _ REQ 命令由轮询设备发送，用于选择指定级联级别下完整的 NFCID1。侦听设备收到 SEL _ REQ 命令后将会响应 SEL _ RES 消息。

SEL _ REQ 命令格式如图 4-20 所示。其中，SEL _ CMD 用于指示 NFCID1 的级联级别。字节 2 为 70h，表示 SEL _ REQ 的消息长度；字节 3～6 表示该级联级别下的 NFCID1；字节 7 为 BCC。从图中可以看出，SEL _ REQ 实际上是一个 7 字节长的 SDD _ REQ 命令。在冲突检测过程中，轮询设备可能需要发送多次 SDD _ REQ 命令才能将该级联级别下的 NFCID1 识别出来。若轮询设备收到的 SDD _ RES 响应消息中没有发生比特冲突，轮询设备将得到该级联级别下完整的 NFCID1 值。轮询设备通过 SEL _ REQ 命令来选中该 NFCID1，并根据 SEL _ RES 响应消息中的指示确定是否已经得到侦听设备所有级联级别完整的 NFCID1。

Byte 1	Byte 2	Byte 3～6	Byte 7
SEL_CMD	70h	NFCID1 CL*n*	BCC

图 4-20 SEL _ REQ 命令格式

侦听设备收到 SEL _ REQ 命令后，将发送 SEL _ RES 进行响应。SEL _ RES 响应消息的格式如图 4-21 所示。SEL _ RES 响应消息只包含一个字节。b3 称为级联比特，用于表示 NFCID1 是否已经完整。这里所说的完整，是指告诉轮询设备是否获取了所有级联级别的 NFCID1，即是否获取了 4 字节或 7 字节或 10 字节的 NFCID1。如果未获取完整的 NFCID1，轮询设备将继续发送 SDD _ REQ 命令来获取下一个级联级别的 NFCID1。如果 b3 指示 NF-CID1 已经完整，轮询设备就完成了一轮单设备检测，识别出一个侦听设备。如果在冲突检测中发生过冲突，证明周围还有其他侦听设备，轮询设备会发送休眠命令，让已检测出的设备休眠，并继续发送 SDD _ REQ 命令进行冲突检测。已休眠的设备不会再响应 SDD _ REQ 命令。另外，b7 和 b6 用于指示侦听设备支持的特性，如类型 2 标签平台、类型 4A 标签平台、NFC-DEP 协议以及同时支持类型 4A 标签和 NFC-DEP 协议。读者可能会，为什么类型 1 标签平台会在 SENS _ RES 中标识，而类型 2 标签平台、类型 4A 标签平台、NFC-DEP 协议在 SEL _ RES 消息中标识？这是因为尽管类型 1 标签平台是 NFC-A 的子集，但是没有采用 NFC-A 的冲突检测方法，而类型 2 标签平台、类型 4A 标签平台以及采用 NFC-DEP 协议的设备都采用了 NFC-A 冲突检测方法，所以需要将配置为类型 1 标签平台的侦听设备区分出来。

4.2.2.6 SLP_ REQ

SLP _ REQ 命令由轮询设备发送给侦听设备，让侦听设备进入睡眠状态。SLP _ REQ 命令的消息格式如图 4-22 所示。当轮询设备识别出侦听设备，即获取侦听设备完整的 NF-CID1 后，会发送 SLP _ REQ 命令给侦听设备。由于被检测出 NFCID1 的侦听设备会处于某

b8	b7	b6	b5	b4	b3	b2	b1	描述
×								任何值
	×	×						如果b3设置为1 - b7b6为任意值 如果b3设置为0 - b7b6=00，类型2标签平台 - b7b6=01，类型4A标签平台 - b7b6=10，支持NFC-DEP协议 - b7b6=11，支持NFC-DEP和类型4A标签平台
			×	×				任何值
					×			级联比特 - 1：NFCID1不完整 - 0：NFCID1完整
						×	×	任何值

图 4-21　SEL _ RES 响应消息格式

个特定的状态，在该状态下接收 SLP _ REQ 命令，其他未被检测出 NFCID1 的侦听设备会忽略该命令。关于侦听设备的状态将在 4.2.3.2 节中介绍。侦听设备睡眠后，轮询设备会继续冲突检测过程，来识别其他侦听设备；已睡眠的侦听设备将不会响应 SDD _ REQ 消息。

Byte 1	Byte 2
50h	00h

图 4-22　SLP _ REQ 命令消息格式

侦听设备收到 SLP _ REQ 命令后进入睡眠模式，并且不会向轮询设备发送响应消息。轮询设备可以通过 ALL _ REQ 命令将其唤醒。

4.2.3　冲突检测

4.2.3.1　冲突检测原理

轮询设备周围可能存在多个侦听设备。当轮询设备发送 SENS _ REQ 命令进行探测时，支持 NFC-A 技术的侦听设备都会响应 SENS _ RES 消息。如果有两个或两个以上的侦听设备同时响应，就会产生冲突，数据之间将相互干扰，轮询设备无法识别出每个侦听设备的 NFCID1。这里假设有两个支持 NFC-A 的侦听设备 A 和 B，其 NFCID1 分别为 00110011 和 00100100（为简便起见，假设 NFCID1 为一个字节）。当轮询设备发送 SDD _ REQ 命令后，由于 SDD _ REQ 只包含 SEL _ CMD 和 SEL _ PAR 两个字节，没有数据位，因此 A 和 B 都会响应 SDD _ RES 消息。两个 SENS _ RES 消息在第 4 位、第 6 位、第 7 位和第 8 位上产生冲突，轮询设备无法识别出 A 和 B 的 NFCID1，如图 4-23 所示。

为了解决冲突的问题，在 NFC-A 中采用基于比特的防冲突算法，其基本思路为：将一个侦听设备检测出来之后，再检测另一个侦听设备。每一次发现冲突，将会在冲突的位上添 1，然后继续检测，直到检测出一个侦听设备。然后，令该设备休眠，不再参与接下来的冲突检测过程。在所有的级联级别中重复以上过程，直到检测出所有的侦听设备。依然以图 4-23为例，轮询设备收到两个 SENS _ RES 消息后，发现在第 4 位发生了冲突，于是轮询设备在 SENS _ REQ 消息中的数据位中将第 4 位设置为 1，然后发送，如图 4-24 所示。在第二轮检测中，SDD _ REQ 的数据位的前三位即为检测出的无冲突位（001），而在第四位的冲

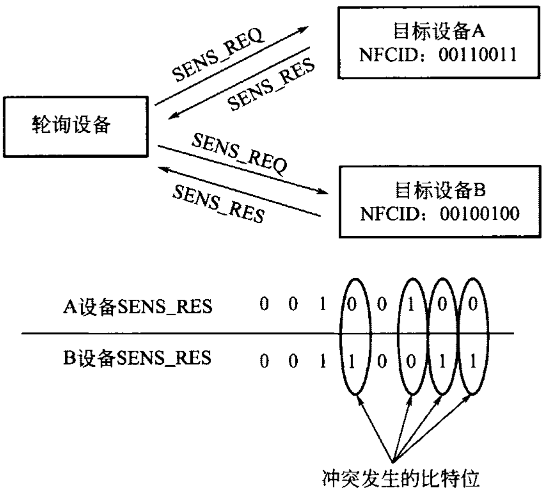


图 4-23 NFC-A 比特冲突举例

突位上设置为 1。设备 A 和设备 B 收到该 SDD _ REQ 消息后，对比 SDD _ REQ 中的数据位与自己的 NFCID 的前 4 位是否一致。本例中与 B 设备匹配，因此 B 设备将向轮询设备回复 SDD _ RES 响应消息，响应消息中包含 NFCID 的后 4 位数据。而 A 设备不匹配，因此不会向轮询设备回复响应消息。这样，轮询设备将获取 B 设备的完整 NFCID1 值。接着，轮询设备向 B 设备发送 SLP _ REQ 命令，令 B 设备休眠，之后再次发送 SDD _ REQ 命令，重新进行冲突检测。SDD _ REQ 命令中不包含数据位。A 设备收到后，将回复 SDD _ RES 消息，其中包含自己的 NFCID1。由于 B 设备在休眠，不会响应，因此没有发生冲突，轮询设备就获取了 A 设备的 NFCID1。

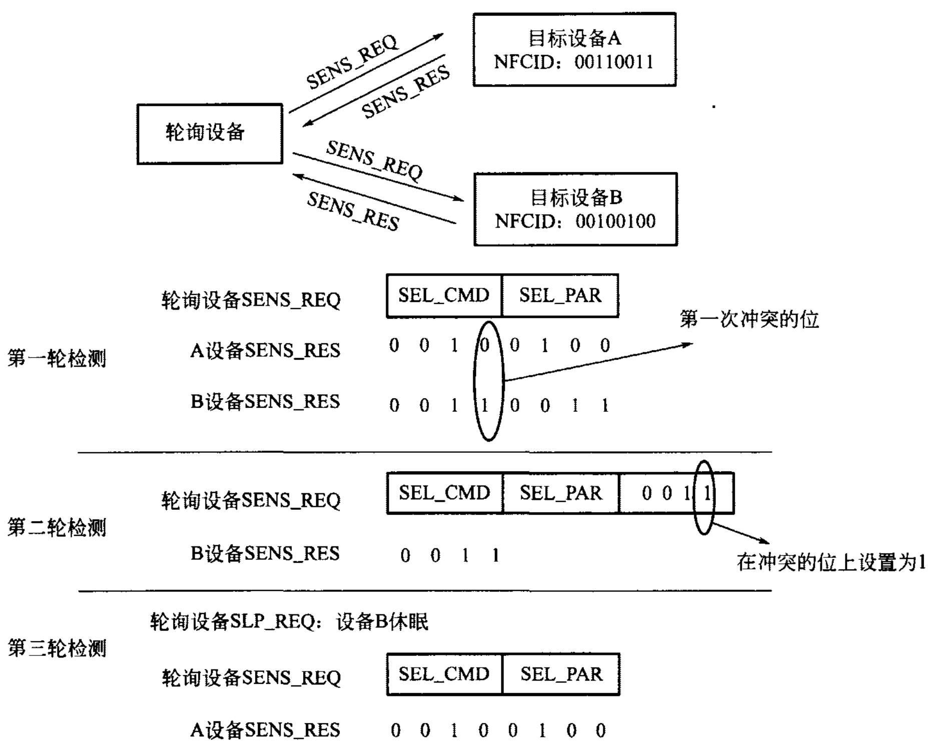


图 4-24 NFC-A 冲突检测举例

冲突检测可以识别出所有的侦听设备，但是选择和哪一个设备通信，并不在标准规范中定义。这往往取决于上层应用或实现的策略。

4.2.3.2 轮询设备冲突检测流程

轮询设备冲突检测流程如图 4-25 所示，操作步骤如下所述。

步骤 1：轮询设备发送 SENS _ REQ 或 ALL _ REQ 命令后，收到侦听设备回复的 SENS _ RES 消息。轮询设备检查 SENS _ RES 中的 b5～b1，判断侦听设备是否支持冲突检测。如果支持，继续判断是否配置为类型 1 标签平台。如果是，执行类型 1 标签平台冲突检测流程（在类型 1 标签平台的小节中介绍），否则执行步骤 2。如果不支持冲突检测，执行结束。

步骤 2：轮询设备选择级联级别 1。

步骤 3：轮询设备设置 SDD _ REQ 命令中的 SEL _ CMD 与 SEL _ PAR 字节，并发送 SDD _ REQ 消息。

步骤 4：轮询设备接收 SDD _ RES 消息并判断是否产生冲突。如果是，执行步骤 5，否则执行步骤 9。

步骤 5：轮询设备检查配置参数，判断是否进行冲突设备的识别（根据配置参数，轮询设备也可以只检测是否发生冲突，但不去识别设备）。如果是，执行步骤 6，否则执行结束。

步骤 6：轮询设备识别出有效的数据位（即被识别出的没有发生比特冲突的数据位）。

步骤 7：轮询设备设置 SEL _ PAR，将有效数据位包含在 SEL _ PAR 之后，并将冲突位设置为 1，然后发送 SDD _ REQ 命令。

步骤 8：轮询设备接收 SDD _ RES 消息，判断是否产生冲突。如果产生冲突，执行步骤 6，否则执行步骤 9。

步骤 9：轮询设备发送 SEL _ REQ 命令。

步骤 10：轮询设备接收 SEL _ RES 消息，并判断 NFCID1 是否完整。如果完整，执行步骤 11；否则增加级联级别，并执行步骤 3。

步骤 11：轮询设备存储 NFCID1。

步骤 12：轮询设备检查配置参数，判断是否继续进行冲突检测。如果是，则发送 SLP _ REQ，令目标设备休眠，并发送 SENS _ REQ 重新开始检测；否则执行结束。

4.2.3.3 侦听设备冲突检测流程

侦听设备冲突检测流程如图 4-26 所示。

① 处于 IDLE 状态的侦听设备收到 ALL _ REQ/SENS _ REQ 命令后进入 READY _ A 状态。

② 在 READY _ A 状态下，侦听设备进入级联级别 1 的冲突检测过程，只能响应 SDD _ REQ 和 SEL _ REQ 命令。接收到任何其他命令，都会使侦听设备返回 IDLE 状态。如果在级联级别 1 中遇到了比特冲突，侦听设备会一直处于该状态进行冲突检测，直到轮询设备识别出该级联级别 1 的 NFCID1。如果侦听设备的 NFCID1 为 4 字节，那么收到 SEL _ REQ 命令后，侦听设备进入 ACTIVE _ A 状态；否则，侦听设备进入 READY _ A′状态，进行级联级别 2 的冲突检测。

③ READY _ A′与 READY _ A 类似，只不过是在级联级别 2 上进行冲突检测，直到轮询设备识别出该级联级别的 NFCID1，侦听设备将会进入下一个状态。如果侦听设备的 NFCID1为 7 字节，收到 SEL _ REQ 命令后进入 ACTIVE _ A 状态，否则进入 READY _ A″状态。

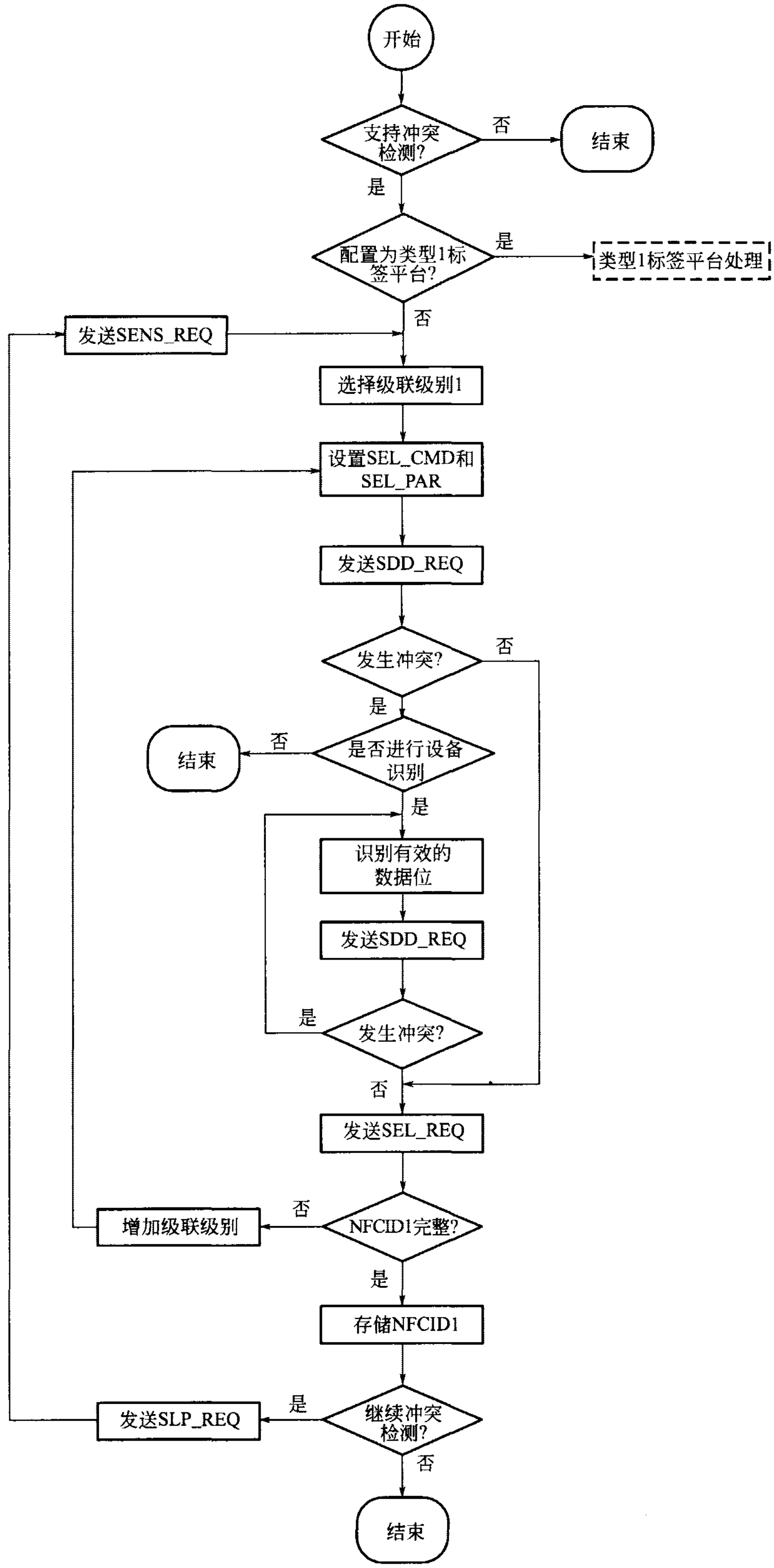


图 4-25 轮询设备冲突检测流程

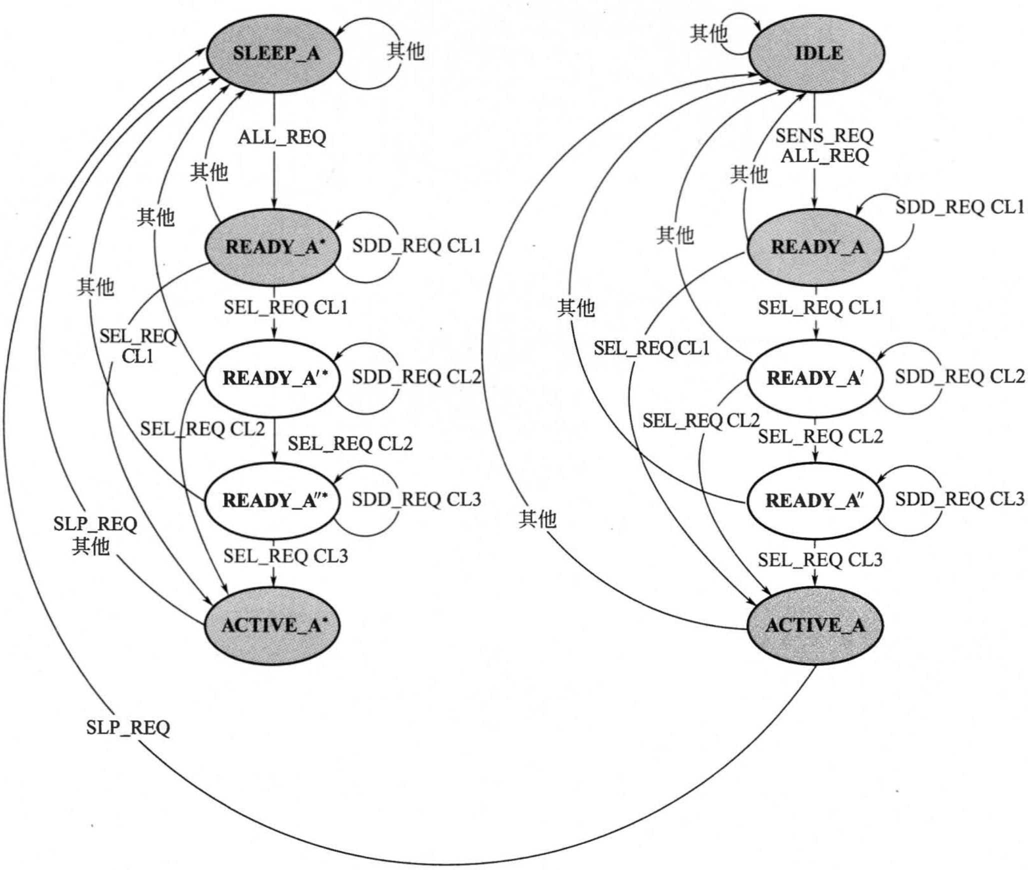


图 4-26 侦听设备冲突检测流程

④ READY _ A″是级联级别 3 的冲突检测状态。当轮询设备识别出该级联级别的 NFCID1 时，侦听设备将进入 ACTIVE _ A 状态。

⑤ 在 ACTIVE _ A 状态下，侦听设备的 NFCID1 已经完整地被轮询设备获得。轮询设备如果要继续进行冲突检测，将会给侦听设备发送 SLP _ REQ 命令。若侦听设备在该状态下收到 SLP _ REQ 命令，将进入 SLEEP _ A 状态。如果在该状态下收到激活命令，侦听设备将进入设备激活的状态。设备激活将在接下来的小节中介绍。

⑥ 在 SLEEP _ A 状态下，侦听设备只能响应 ALL _ REQ 命令。该状态下的侦听设备收到 ALL _ REQ 命令后，进入 READY _ A* 状态。READY _ A* /READY _ A′* / READY _ A″* 与 READY _ A/READY _ A′/READY _ A″状态的跃迁条件一致，区别在于是从 IDLE 状态触发，还是从 SLEEP _ A 状态触发。

这里需要注意的是，除了最后一个被检测出来的侦听设备外，在冲突检测过程中被检测出来的其他侦听设备都会被设置为休眠状态，即进入 SLEEP _ A 状态。如果需要唤醒和激活其中某一个处于 SLEEP _ A 状态下的侦听设备，即从 SLEEP _ A 状态到 ACTIVE _ A* 状态，同样需要经过类似的冲突检测过程。由于轮询设备知道要被激活的侦听设备的 NFCID1，因此只需要根据 NFCID1 的长度，在相应的级联级别下发送 SEL _ REQ 命令即可。但是，由于轮询设备周围可能依然存在未处于休眠的侦听设备，因此有冲突的可能性，所以在 READY _ A* 、READY _ A′* 和 READY _ A″* 状态下，也会有交互 SDD _ REQ 命令的可能性。

4.3 NFC-B

4.3.1 帧格式

NFC-B 帧由字符组成，一个字符由一个起始位、一个结束位以及 8 个数据位组成。起始位为逻辑 “0”，结束位为逻辑 “1”。起始位、结束位和每个数据位都有 1bit 的持续时间，称为 bd（bit duration)。图 4-27 描述了一个字符的格式，一个字符有 10bit，字符的长度为 10bd。

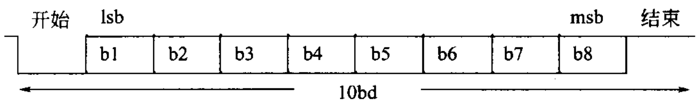


图 4-27 NFC-B 字符

NFC-B 的帧格式如图 4-28 所示，它由字符组成。

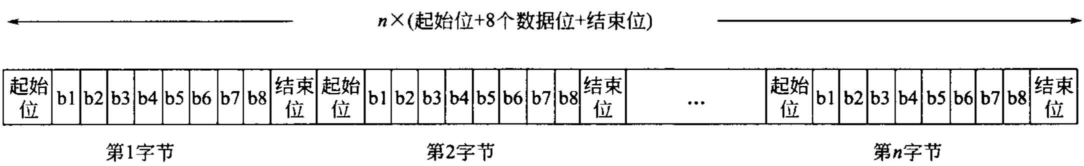


图 4-28 NFC-B 帧格式

NFC-B 的数据格式与 NFC-A 一样，都没有 SOD，如图 4-29 所示。

数据(命令或响应)				[EOD]	
Byte 1	Byte 2	…	Byte *n*	[CRC_A1]	[CRC_A2]

图 4-29 NFC-B 数据格式

4.3.2 命令

NFC-B 的命令集如表 4-3 所示。

表 4-3 NFC-B 命令集

命　令	响　应
ALLB_REQ,SENSB_REQ	SENSB_RES
SLOT_MARKER(在轮询模式下是可选的)	SENSB_RES
SLPB_REQ	SLPB_RES

4.3.2.1 ALLB_ REQ/SENSB_ REQ

ALLB _ REQ 和 SENSB _ REQ 命令用于技术发现过程，由轮询设备发送来探测支持 NFC-B 技术的侦听设备，如图 4-30 所示。ALLB _ REQ 和 SENSB _ REQ 的主要区别在于 ALLB _ REQ 用于唤醒已经被检测出，但是处于睡眠的侦听设备；SENS _ REQ 用于探测还未被检测出的侦听设备。ALLB _ REQ 和 SENSB _ REQ 的格式一样，二者是通过 PARAM 中的 b4 来区分的。

Byte 1	Byte 2	Byte 3
05h	AFI	PARAM

图 4-30 ALLB _ REQ 消息格式

AFI（Application Family Identifier，应用族指示符）用于轮询设备标识选中的应用领域，比如若 AFI 为 00，表示所有的应用领域；10 表示交通领域，20 表示金融领域。在 Digital 规范中，轮询设备发送的 ALLB _ REQ/SENSB _ REQ 中的 AFI 被设置为 00，表示支持所有应用领域。另外，Digital 规范规定，侦听设备需要支持 AFI 为非 00 的情况。

PARAM 长度为 1 字节，用于标识轮询设备所支持的特性，其编码格式如图 4-31 所示。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0							预留
		×						0：不支持高级协议特性 1：支持高级协议特性
			×					0：支持扩展的SENSB_RES 1：不支持扩展的SENSB_RES
				×				0：SENSB_REQ 1：ALLB_REQ
					×	×	×	时隙数*N*

图 4-31 PARAM 编码格式

b6 用于描述轮询设备是否支持高级协议特性，如更高的数据传输速率等。

b5 表示轮询设备是否支持扩展的 SENSB _ RES 消息。扩展的 SENSB _ RES 消息是指在该响应消息中有一个可选的扩展 SENSB _ RES 字节，其中包含 SFGI（Start-up Frame Guard Time Integer）信息，该信息是用于描述轮询设备与侦听设备之间的帧时延。

b3	b2	b1	描述
0	0	0	*N*=1
0	0	1	*N*=2
0	1	0	*N*=4
0	1	1	*N*=8
1	0	0	*N*=16
1	0	1	预留
1	1	×	预留

图 4-32 时隙数 *N* 编码格式

b3～b1 用于描述冲突检测时隙数 *N*。NFC-B 的冲突检测过程与 NFC-A 不同，NFC-A 是基于比特的冲突检测，而 NFC-B 基于时隙，即采用时分复用的方法来避免冲突。ALLB _ REQ/SENSB _ REQ 中指定了时隙的个数 *N*。该时隙数 *N* 是指轮询设备能够接收 SENSB _ RES 响应消息的时间窗口的个数。侦听设备将根据时隙数，随机选择一个时隙窗口发送 SENSB _ RES 响应消息，从而最大程度地减少冲突的概率。时隙数 *N* 的编码格式如图 4-32 所示。如果 *N* 设置为 1，表示轮询设备只有一个时间窗口来接收 SENSB _ RES 消息；如果存在多个侦听设备，都会选择该时间窗口发送 SENSB _ RES 响应消息，将必定产生冲突，因此 $N=1$，表明轮询设备不支持冲突检测过程。

b4 用于标识该命令是 ALLB _ REQ 还是 SENSB _ REQ。

4.3.2.2 SENSB_ RES

SENSB _ RES 由侦听设备发送，用于响应轮询设备发送的 SENSB _ REQ/ALLB _ REQ 命令，其格式如图 4-33 所示。

Byte 1	Byte 2～5	Byte 6～9	Byte 10～12或Byte 13
50h	NFCID0	应用数据	协议信息

图 4-33 SENSB_RES 消息格式

NFC-B 的 NFCID 称为 NFCID0，由 4 个字节组成。NFCID0 用于 NFC-B 冲突检测过程中，唯一地标识一个侦听设备。NFCID0 可以是固定值，也可以是设备从关闭（Power-off）状态跃迁到空闲（IDLE）状态的过程中动态生成的。

SENSB_RES 响应消息中的应用数据用于告诉轮询设备，在侦听设备上安装了哪些应用。

协议信息为第 10 字节到第 12 字节，如果包含了前面提到的扩展 SENSB_RES 字节，协议信息就是从第 10 字节到第 13 字节。协议信息用于描述侦听设备支持的一些参数，如所支持的传输速率、最大帧长度、帧等待时间（从收到轮询帧到发送侦听帧的间隔时间）、SFGI 等。

4.3.2.3 SLOT_ MARKER

SLOT_MARKER 用于在冲突检测过程中通知侦听设备某个响应时隙的开始。该消息是可选的。SLOT_MARKER 命令的格式如图 4-34 所示。

Byte 1
AP*n*

图 4-34 SLOT_MARKER 消息格式

SLOT_MARKER 只有一个字节，其中 b4～b1 被设置为固定值 0101，b8～b5 编码格式如图 4-35 所示。

b8	b7	b6	b5	描述
0	0	0	0	不允许
0	0	0	1	时隙2
0	0	1	0	时隙3
0	0	1	1	时隙4
…	…	…	…	…
1	1	1	0	时隙15
1	1	1	1	时隙16

图 4-35 SLOT_MARKER b8～b5 编码格式

Byte 1	Byte 2～5
50h	NFCID0

图 4-36 SLPB_REQ 命令格式

Byte 1
00h

图 4-37 SLPB_RES 消息格式

侦听设备收到 SLOT_MARKER 命令后，会决定是否在该时隙上回复 SENSB_RES 响应消息。

4.3.2.4 SLPB_ REQ/SLPB_ RES

轮询设备发送 SLPB_REQ 命令，将指定的 NFCID0 侦听设备设置为睡眠状态。SLPB_REQ 命令消息格式如图 4-36 所示。

收到 SLPB_REQ 命令的侦听设备将响应 SLPB_RES 消息，其消息格式如图 4-37 所示。

4.4 NFC-F

4.4.1 帧格式

与 NFC-B 类似，NFC-F 帧由字符组成，一个字符包含 8 个数据位，但是没有起始位、结束位和校验位，如图 4-38 所示。字符之间没有任何间隔，以字符串的形式连续传输。

msb							lsb
b8	b7	b6	b5	b4	b3	b2	b1

图 4-38　NFC-F 字符

NFC-F 的帧由字符组成，以 SOF 开始，如图 4-39 所示。

SOF	数据(由字符组成)

图 4-39　NFC-F 帧格式

NFC-F 的数据格式如图 4-40 所示，由 SOD、负载（Payload）和 EOD 组成。

SOD	负载				EOD	
LEN	Byte 1	Byte 2	…	Byte *n*	CRC_F1	CRC_F2

图 4-40　NFC-F 数据格式

4.4.2 命令

NFC-F 的命令集只有 SENSF _ REQ 命令和响应消息 SENSF _ RES，如表 4-4 所示。

表 4-4　NFC-F 命令集

命　令	响　应
SENSF_REQ	SENSF_RES

4.4.2.1 SENSF_ REQ

轮询设备发送 SENSF _ REQ 来探测周围支持 NFC-F 技术的侦听设备。SENSF _ REQ 命令的格式如图 4-41 所示。

SC（System Code，系统代码）包含了关于被轮询设备的一些信息请求，如轮询设备所支持的技术子集等。在 Digital 规范中，要求 SC 设置为 FFFFh。

RC（Request Code，请求代码）用于从侦听设备处获取更多的信息，这些信息通过 SENSF _ RES 响应消息携带。RC 的具体含义如图 4-42 所示。

Byte 1	Byte 2～3	Byte 4	Byte 5
00h	SC	RC	TSN

图 4-41　SENSF _ REQ 命令格式

值	描述
00h	无SC请求信息
01h	请求SC信息
02h	支持高级协议特性
其他值	预留

图 4-42　RC 的含义

如果轮询设备被配置为支持 NFC-DEP 的设备，RC 必须设置成 00h，即如果轮询设备采用点对点通信，轮询设备无需请求 SC 的信息。如果轮询设备被配置为类型 3 的标签平台，RC 必须被设置为 00h 或 01h，即如果轮询设备为类型 3 标签的读卡器，轮询设备可以请求或不请求类型 3 标签的 SC 信息。在 Digital 1.0 规范中暂不支持高级协议特性。

TSN（Time Slot Number，时隙数）用于在冲突检测过程中减少冲突的概率。NFC-F 采用了基于时隙的冲突检测方式，即侦听设备在不同的时间窗口返回响应消息，从而避免冲突的概率。轮询设备在 SENSF _ REQ 命令中设置 TSN 的值，用于指示有多少个时隙可用。每一个侦听设备收到 SENSF _ REQ 命令后，在 TSN 的时隙范围内随机选择一个时隙回复响应消息。TSN 编码格式如图 4-43 所示。

TSN	00h	01h	03h	07h	0Fh	其他值
时隙数	1	2	4	8	16	预留
SENSF_RES可选择的时隙	时隙0	时隙0 时隙1	时隙0 … 时隙3	时隙0 … 时隙7	时隙0 … 时隙15	

图 4-43 TSN 编码格式

如果轮询设备不支持冲突检测，可以将 TSN 设置为 00h，那么所有的侦听设备被强制在第一个时隙发送响应消息。图 4-44 举例说明了 NFC-F 的冲突检测机制。轮询设备在 SENSF _ REQ 中设置 TSN 为 4，即有 4 个时间窗口可选。侦听设备 D 和 B 选择了不同的时隙发送响应消息，因此轮询设备可以将 B 和 D 区分开来。侦听设备 A 和 C 选择了同一个时隙发送响应消息，发生了冲突，因此轮询设备无法区分设备 A 和 C。在 Digital 规范中，时隙从 0 开始计数。冲突检测的目的就是识别出每一个侦听设备的 NFCID 值。

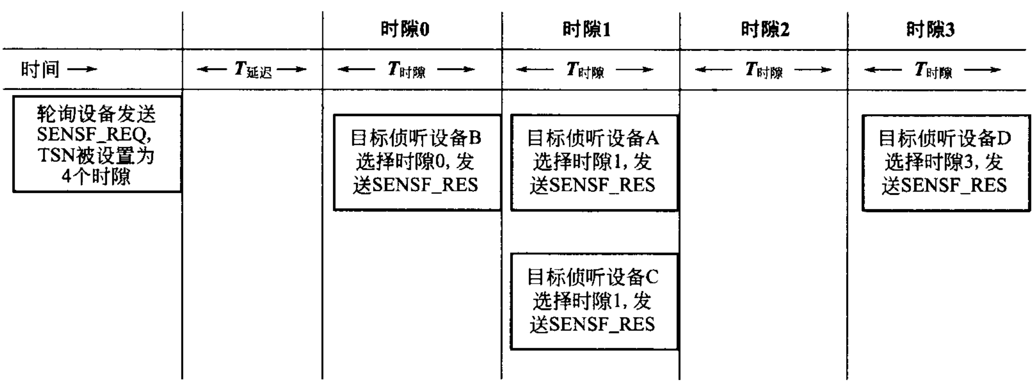


图 4-44 NFC-F 冲突检测举例

4.4.2.2 SENSF_ RES

SENSF _ RES 响应消息格式如图 4-45 所示。

NFC-F 中的 NFCID 称为 NFCID2，长度为 8 字节，编码格式如图 4-46 所示。NFCID2 用于 NFC-F 冲突检测过程中唯一地标识一个侦听设备。NFCID2 的第一个字节和第二个字

Byte 1	Byte 2～9	Byte 10～11	Byte 12～14	Byte 15	Byte 16	Byte 17	[Byte 18～19]
01h	NFCID2	PAD0	PAD1	$MRTI_{检测}$	$MRTI_{更新}$	PAD2	[RD]

图 4-45 SENSF _ RES 响应消息格式

节用于表示该设备支持的一些特性，如 Byte1 和 Byte2 取值为 01h 和 FEh，表示该目标设备支持 NFC-DEP 协议，Byte3～Byte8 可以动态随机生成。

Byte 1	Byte 2	描述
01h	FEh	支持NFC-DEP，Byte3～Byte8随机生成
02h	FEh	支持类型3标签平台，Byte3～Byte8随机生成
其他值		支持类型3标签平台，Byte1～Byte8为私有的

图 4-46 NFCID2 编码格式

对于 Byte10～Byte11 的 PAD0，在 Digital 规范中并未使用该值。Digital 规范兼容 Felica JIS X6319-4 规范，因此在消息格式上完全一致，但是在消息中某些字节的使用上，Digital 规范并未完全采用 JIS X6319-4 的用法，这类字节一般都是私有的。

PAD1、$MRTI_{检测}$、$MRTI_{更新}$、PAD2 只用于类型 3 标签平台。如果采用 NFC-DEP 协议，这些字节将不被使用。这些字节将在 4.7 节介绍。

对于 RD（Request Data），如果 SENSF_REQ 中的 RC 字段设置为 00h，SENSF_RES 中将不包含 RD。如果 SENSF_REQ 中的 RC 字段被设置为 01h 或 02h，并且侦听设备配置为类型 3 的标签平台，那么 SENSF_RES 中包括 2 字节 RD 字段。RD 包含 2 个字节，第 1 个字节设置为 00h，第 2 个字节表示所支持的比特速率，编码格式如图 4-47 所示。

b8	b7	b6	b5	b4	b3	b2	描述
1							自动速率选择
	0						预留
		0					预留
			0				预留
				×			速率为1695Kbps，目前不支持
					×		速率为848Kbps
						×	速率为424Kbps
							× 速率为212Kbps

图 4-47 RD 编码格式

4.4.3 冲突检测

4.4.3.1 轮询设备冲突检测流程

轮询设备冲突检测流程如图 4-48 所示。

步骤 1：轮询设备发送 SENSF_REQ 命令探测周围设备。

步骤 2：支持 NFC-F 技术的侦听设备收到 SENSF_REQ 命令后，根据 SENSF_REQ 命令中的 TSN，随机选择一个时隙响应 SENSF_RES 消息。

步骤 3：轮询设备在每一个时隙上侦听，收集有效的 SENSF_RES 消息，并存储在响应消息列表中。

步骤 4：轮询设备比较有效的 SENSF _ RES 消息数量与事先配置的所支持的目标设备数量的参数。如果前者小于后者，执行步骤 5，否则执行步骤 10。

步骤 5：轮询设备设置 TSN 为最大时隙数 16，设置 RC 为“无 SC 请求信息”，SC 为 FFFFh，并发送 SENSF _ REQ 命令。

步骤 6：轮询设备接收侦听设备响应的 SENSF _ RES 消息，并对比这些响应消息是否在响应消息列表中。如果不在，将其存入响应消息列表。

步骤 7：轮询设备检查 SENSF _ RES 消息中 NFCID2 的字节 1 和字节 2，判断是否有目标侦听设备支持 NFC-DEP 协议。如果有，执行步骤 8，否则执行步骤 10。

步骤 8：轮询设备设置 TSN 为最大时隙数 16，设置 RC 为“请求 SC 信息”，SC 为 FFFFh，并发送 SENSF _ REQ 命令。

步骤 9：轮询设备接收目标侦听设备响应的 SENSF _ RES 消息，并对比这些响应消息是否在响应消息列表中。如果不在，将其存入响应消息列表。

步骤 10：轮询设备更新 NFCID2 列表，完成冲突检测过程。

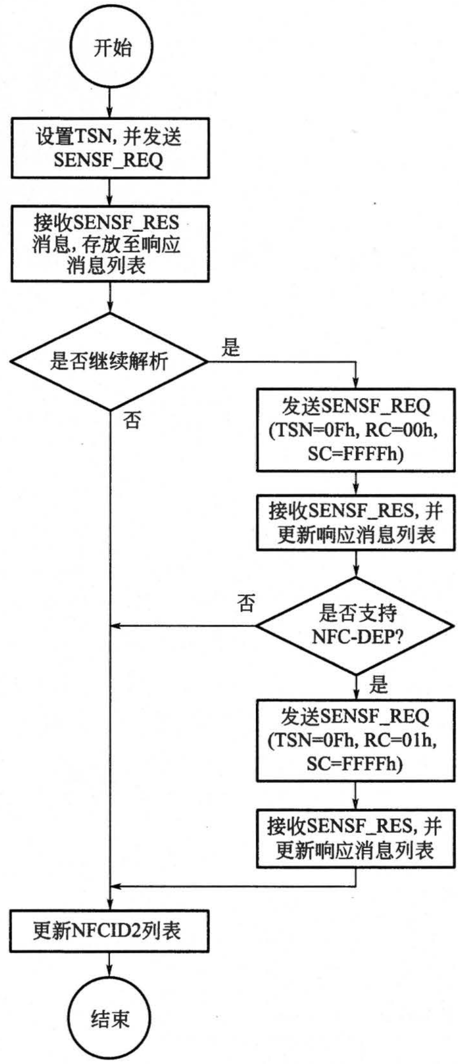


图 4-48 NFC-F 轮询设备冲突检测流程

4.4.3.2 侦听设备冲突检测流程

侦听设备冲突检测状态转移如图 4-49 所示。

① 侦听设备进入 IDLE 状态，等待 SENSF _ REQ 命令到来。

② 侦听设备收到 SENSF _ REQ 命令后，进入 READY _ F 状态。在该状态下，目标侦听设备接收 NFC-DEP 的设备激活命令，或者是类型 3 标签平台的 POLLING、CHECK 或 UPDATE 命令。

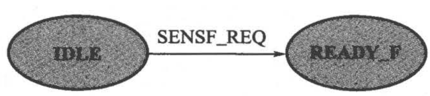


图 4-49 NFC-F 侦听设备冲突检测状态转移

4.5 类型 1 标签平台

4.5.1 封装格式

类型 1 标签平台是 NFC-A 技术的一个子集。在侦听模式下，其帧格式与 NFC-A 的标

准帧一致；而在轮询模式下的帧格式与 NFC-A 标准帧略有不同，没有加入校验位，如图 4-50所示。

	lsb						msb	
SOF	b1	b2	b3	b4	b5	b6	b7	b8	EOF

图 4-50 类型 1 标签平台轮询模式帧格式

类型 1 标签平台的数据格式如图 4-51 所示，与 NFC-A 的数据格式一致，区别在于类型 1 标签平台采用了 NFC-B 技术的循环校验码技术 CRC _ B1 和 CRC _ B2。

数据(命令或响应)				[EOD]	
Byte 1	Byte 2	...	Byte *n*	[CRC_B1]	[CRC_B2]

图 4-51 类型 1 标签平台数据格式

4.5.2 命令

类型 1 标签平台用于支持类型 1 标签的读写器或卡模拟设备，也就是说，配置为标签 1 平台的轮询设备作为类型 1 标签的读写器，配置为类型 1 标签平台的侦听设备可以模拟成类型 1 标签。类型 1 标签平台定义了命令和响应来支持类型 1 标签读写器与类型 1 标签或类型 1 标签的卡模拟设备进行通信，如表 4-5 所示。

表 4-5 类型 1 标签命令及响应

命　令	响　应	命　令	响　应
RID Command	RID Response	RSEG Command	RSEG Response
RALL Command	RALL Response	READ8 Command	READ8 Response
READ Command	READ Response	WRITE-E8 Command	WRITE-E8 Response
WRITE-E Command	WRITE-E Response	WRITE-NE8 Command	WRITE-NE8 Response
WRITE-NE Command	WRITE-NE Response		

在命令响应列表中，除了 RID 命令/RID 响应之外，其他命令响应将在第 6 章标签操作规范中介绍。这些命令以不同的方式进行读写操作，如 RALL 命令，用于读取标签中所有的内容，而 READ8 命令只读取指定的 8 字节。本节主要介绍 RID 命令。该命令用于类型 1 标签平台的冲突检测及设备激活过程。RID 命令消息格式如图 4-52 所示。

Byte 1	**Byte 2**	**Byte 3**	**Byte 4～7**
78h	ADD	DATA	UID-echo

图 4-52 RID 命令消息格式

RID 命令用于获取类型 1 标签或模拟成类型 1 标签的侦听设备的唯一标识符以及所支持协议的信息。其中，字节 2～7 的 ADD、DATA 和 UID-echo 在 Digital 规范中都没有使用，可以被设置为任意值，因此这里不过多介绍。

RID 响应用于类型 1 标签或模拟成类型 1 标签的侦听设备响应轮询设备发送的 RID 命令，其响应消息格式如图 4-53 所示。

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
HR0	HR1	UID0	UID1	UID2	UID3

图 4-53 RID 响应消息格式

其中，字节 1 和字节 2 的 HR0 和 HR1（Header ROM byte）是类型 1 标签 ROM 中最开始的两个固定字节，用于指示该标签是否为类型 1 标签，是否支持 NDEF 消息格式等信息。这两个字节不会被其他读写命令寻址到，只能通过 RID 命令获取。字节 3～6 的 UID0～UID3 用于唯一地标识类型 1 标签平台。这里的 UID0～UID3 相当于 NFC-A 中的 4 字节 NFCID1。

4.5.3 冲突检测

类型 1 标签平台是 NFC-A 技术的子集，但它有自己的冲突检测流程，参照图 4-25 中的虚线部分“类型 1 标签平台处理”，其冲突检测流程如图 4-54 所示。

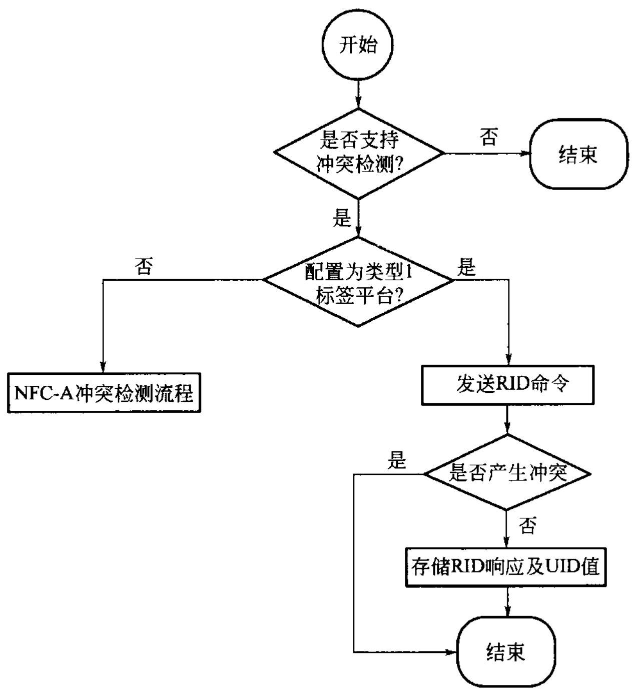


图 4-54 类型 1 标签平台冲突检测流程

步骤 1：轮询设备发送 SENS_REQ 或 ALL_REQ 命令后，收到侦听设备回复的 SENS_RES 消息，轮询设备检查 SENS_RES 中的 b5～b1，确认侦听设备是否支持冲突检测。如果支持，继续判断是否配置为类型 1 标签平台。如果是，执行步骤 2，否则执行 NFC-A 的冲突检测流程。如果不支持冲突检测，执行结束。

步骤 2：轮询设备发送 RID 命令，并接收 RID 响应。

步骤 3：轮询设备判断 RID 响应消息是否产生冲突。如果没有冲突，执行步骤 4，否则执行结束。

步骤 4：轮询设备存储 RID 响应消息及其中的 UID 值，执行结束。

从类型 1 标签平台的冲突检测过程可以看出，只要产生冲突，冲突检测过程就结束；如果未产生冲突，轮询设备通过 UID 来标识侦听设备。

4.5.4 设备激活

由于类型 1 标签平台并未执行真正的冲突检测，它只支持一个侦听设备，因此不存在实质的设备激活过程。通过 RID 获取到目标侦听设备的 UID 后，就可以通过读写命令与侦听设备通信了。

4.6 类型 2 标签平台

4.6.1 封装格式

类型 2 标签平台是 NFC-A 技术的一个子集，除了 ACK 和 NACK 响应外，其他命令和响应消息都采用 NFC-A 标准帧。ACK 和 NACK 是一个包含 4 个数据位的短帧。类型 2 标签平台的数据封装格式如图 4-55 所示。

数据(命令或响应)				[EOD]	
Byte 1	Byte 2	…	Byte *n*	[CRC_A1]	[CRC_A2]

图 4-55 类型 2 标签平台命令响应封装格式

4.6.2 命令

类型 2 标签平台用于支持类型 2 标签的读写器或卡模拟设备，也就是说，配置为标签 2 平台的轮询设备作为类型 2 标签的读写器，配置为类型 2 标签平台的侦听设备模拟成类型 2 标签。类型 2 标签平台定义了命令和响应来支持类型 2 标签读写器与类型 2 标签或类型 2 标签的卡模拟设备进行通信，如表 4-6 所示。

表 4-6 类型 2 标签平台命令及响应

命 令	响 应
READ Command	READ Response, NACK Response
WRITE Command	ACK Response, NACK Response
SECTOR SELECT Command	ACK Response, NACK Response, Passive ACK Response

READ 和 WRITE 命令用于对类型 2 标签或模拟成类型 2 标签的侦听设备进行读写操作。这两个命令都以块为单位进行读写。如果要读写不同扇区中的块数据，需要通过 SECTOR SELECT 命令选择扇区。类型 2 标签平台的命令和响应将在第 6 章介绍。

4.6.3 设备激活

类型 2 标签平台采用 NFC-A 的冲突检测流程。冲突检测完成后，轮询设备发送 READ 或 WRITE 命令激活侦听设备。

对于轮询设备来说，由于在冲突检测过程中将侦听设备都设置为休眠状态，因此在设备激活时，需要将已经休眠的侦听设备唤醒。唤醒操作通过 ALL _ REQ 命令来完成。类型 2 标签平台设备激活流程中的侦听设备的唤醒步骤与 NFC-A 的另外两个技术子集类型 4A 标签平台和 NFC-DEP 是一样的。对于类型 1 标签平台，只支持一个侦听设备，因此不存在设备唤醒和设备激活步骤。对于类型 2 标签平台来说，其实并没有实质的设备激活过程，只需

要将激活的设备唤醒并选中即可。配置为类型 2 标签平台的轮询设备的设备激活流程如图 4-56所示。

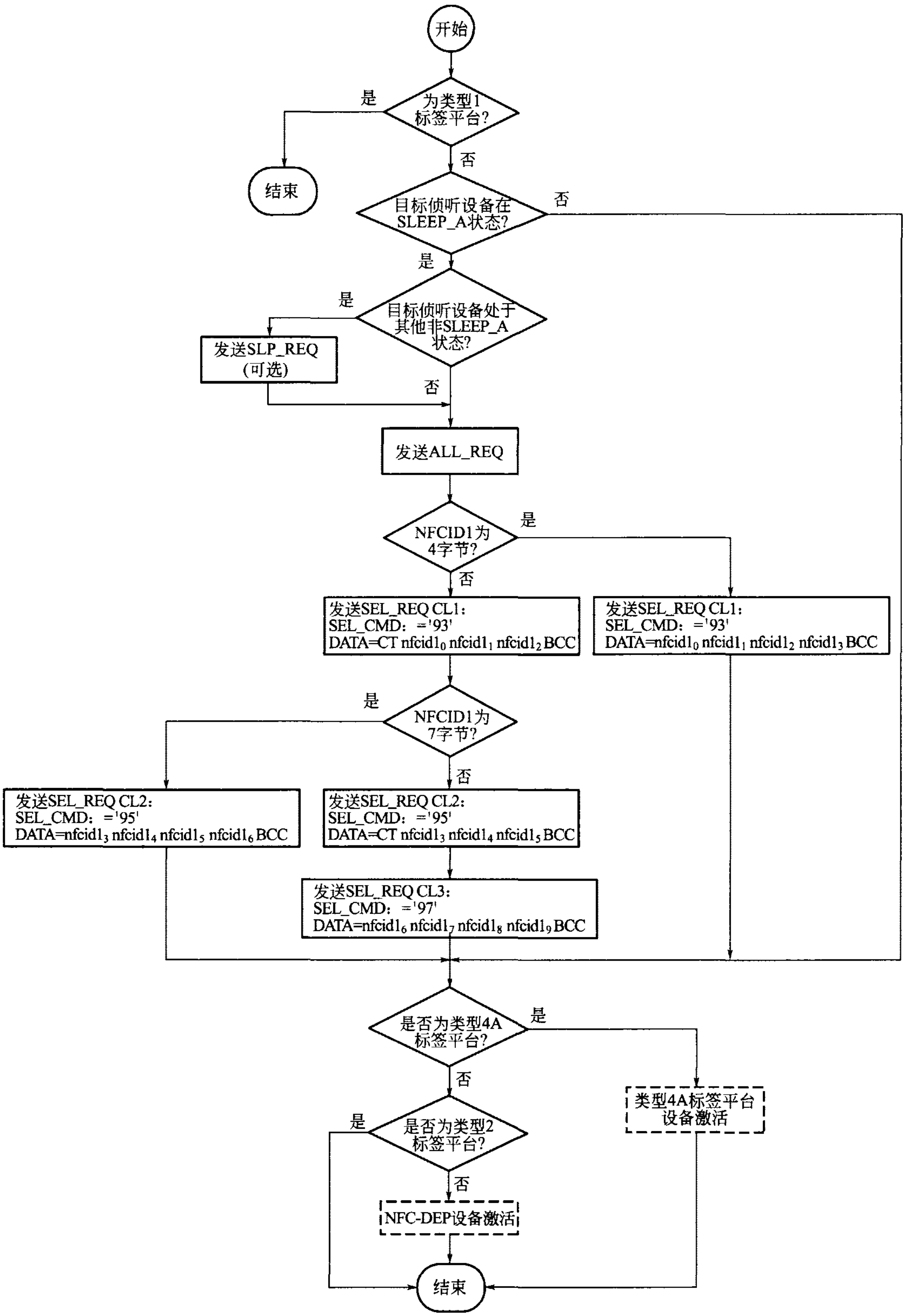


图 4-56　类型 2 标签平台轮询设备激活流程

步骤 1：轮询设备检查需要激活的侦听设备是否为配置成类型 1 标签平台的设备。如果是，执行结束，否则执行步骤 2。由于在冲突检测过程中，轮询设备保存了关于侦听设备的相关信息，因此轮询设备知道需要被激活的侦听设备的信息。至于从休眠设备列表中选择哪个侦听设备来激活，不在 Digital 规范的范围内。一般情况下，由上层应用或服务来决定。

步骤 2：轮询设备判断需要被激活的侦听设备是否在 SLEEP _ A 状态下。如果在 SLEEP _ A 状态下，执行步骤 3，否则执行步骤 6。在 NFC-A 冲突检测完毕后，最后一个被检测出的侦听设备处于非休眠状态，因此如果需要激活的侦听设备为最后检测出的设备，直接执行激活过程；否则，将最后检测出来的侦听设备设置为休眠状态后，将需要激活的侦听设备唤醒，如步骤 3 和步骤 4 所示。

步骤 3：轮询设备判断是否有其他侦听设备还处于非 SLEEP _ A 状态。如果有，发送 SLP _ REQ 命令使其休眠，然后执行步骤 4，否则直接执行步骤 4。这里的其他设备是指轮询设备不准备激活的侦听设备。

步骤 4：轮询设备发送 ALL _ REQ 命令，唤醒所有的侦听设备。发送 ALL _ REQ 命令的目的是让所有休眠的侦听设备进入 READY _ A* 状态。在该状态下，侦听设备可以接收 SDD _ REQ 命令和 SEL _ REQ 命令。

步骤 5：轮询设备判断要激活的侦听设备的 NFCID1 长度，按照 NFCID1 长度发送 SEL _

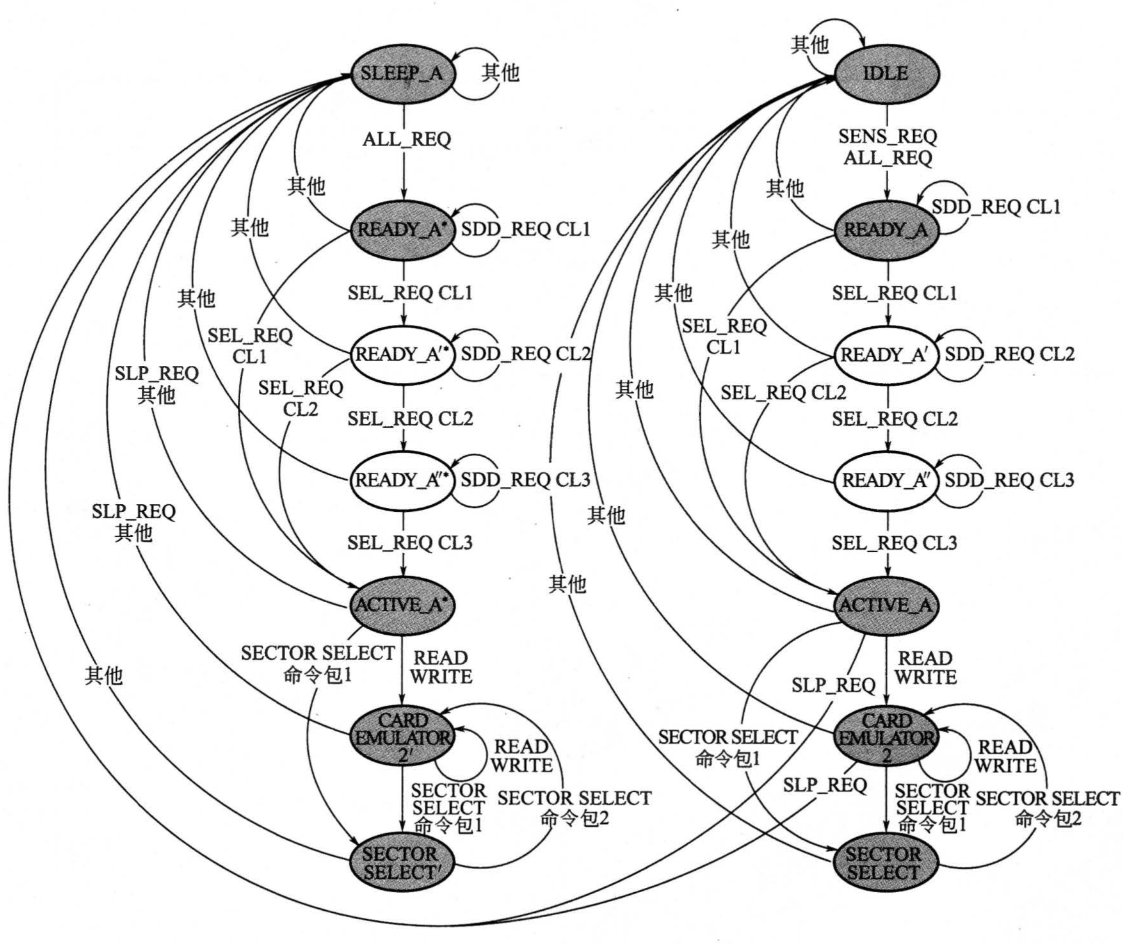


图 4-57 配置为类型 2 标签平台的侦听设备的状态转移图

REQ 命令。由于轮询设备知道要激活的侦听设备的 NFCID1，因此无需重复冲突检测过程。如果 NFCID1 长度为 4 字节，只需发送级联级别 1 的 SEL_REQ 命令；如果 NFCID1 为 7 字节，需要发送两个级联级别的 SEL_REQ 命令来选择；如果 NFCID1 为 10 字节，需要发送三个级联级别的 SEL_REQ 命令。

步骤 6：当轮询设备把要激活的侦听设备选择出来后，判断该侦听设备是否支持类型 4A 标签平台。如果是，执行类型 4A 标签平台的设备激活过程，具体内容将在 4.8 节介绍；否则执行步骤 7。

步骤 7：轮询设备判断要激活的侦听设备是否为类型 2 标签平台。如果是，执行结束，否则执行 NFC-DEP 的设备激活过程，具体内容将在 4.12 节介绍。

侦听设备可以从 ACTIVE_A 状态或 ACTIVE_A* 状态进入 CARD EMULATOR 2 状态。在该状态下，轮询设备发送 READ 或 WRITE 命令进行读写操作。如果收到 SECTOR SELECT 命令包 1，侦听设备进入 SECTOR SELECT 状态。在该状态下不能接收 READ/WRITE 命令，收到 SECTOR SELCET 命令包 2 才能再次进入 CARD EMULATOR 2 状态，并接收读写操作。配置为类型 2 标签平台的侦听设备的状态转移如图 4-57 所示。

4.7 类型 3 标签平台

4.7.1 封装格式

在 Digital 规范中，类型 3 标签平台的侦听模式是可选的，也就是说，NFC 设备模拟成类型 3 的标签这个功能是可选的。类型 3 标签平台采用了 NFC-F 的帧格式及数据封装格式。

4.7.2 命令

类型 3 标签平台用于支持类型 3 标签的读写器或卡模拟设备，也就是说，配置为类型 3 标签平台的轮询设备作为类型 3 标签的读写器，配置为类型 3 标签平台的侦听设备模拟成类型 3 标签。类型 3 标签平台定义了命令和响应来支持类型 3 标签读写器与类型 3 标签或类型 3 标签的卡模拟设备进行通信，如表 4-7 所示。

表 4-7 类型 3 标签平台命令及响应

命令	响应
POLLING Command	POLLING Response
CHECK Command	CHECK Response
UPDATE Command	UPDATE Response

POLLING 命令和响应实际上就是 NFC-F 中介绍的 SENSF_REQ 和 SENSF_RES 消息，只不过在 JIS X6319-4 规范中称为 POLLING 命令和响应。

CHECK 命令用于读取数据，UPDATE 用于写数据。类型 3 标签平台也是按照块进行数据读写，但它支持一次读写多个块的数据。

4.7.3 设备激活

对于配置为类型 3 标签平台的轮询设备来说，没有设备激活过程，在冲突检测过程完毕后，轮询设备直接向配置为类型 3 标签平台的侦听设备或类型 3 标签发送读写命令。对

于侦听设备来说，图 4-49 所示为 NFC-F 的冲突检测状态转移图。基于该状态转移图，图 4-58 描述了设备激活的流程。

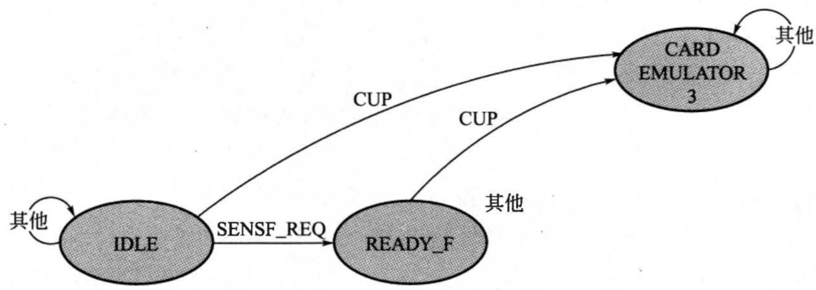


图 4-58　配置为类型 3 标签平台的侦听设备的状态转移图

图中，CUP 是指 CHECK、UPDATE 和 Proprietary。其中，Proprietary 为 Felica 中的私有命令，Digital 规范中没有任何定义。

在 CHECK 和 UPDATE 命令中都包含 “Manufacture ID（IDm）” 的参数，该参数实际上就是标识该设备的 NFCID2，因此在 IDLE 状态下的侦听设备收到 CHECK 或 UPDATE 命令后，如果命令中的 IDm 与自己的 NFCID2 一致，侦听设备就进入 CARD EMULATOR 3 状态。从 READY _ F 状态收到 CUP 命令，同样可以进入 CARD EMULATOR 3 状态。

直接从 IDLE 状态进入 CARD EMULATOR 3 状态的场景一般应用在轮询设备事先知道侦听设备 NFCID2 的情况，因此可以直接读写。从 READY _ F 进入 CARD EMULATOR 3 的场景，需要先获取侦听设备的 NFCID2，之后才能使用 CUP 命令操作。另外，在 READY _ F 状态下，如果侦听设备也支持 NFC-DEP 协议，那么收到 NFC-DEP 的激活命令后，就会进入 ATR _ READY _ F 状态。该状态将在 4.12 节介绍。

4.8 类型 4A 标签平台

4.8.1 封装格式

类型 4A 标签平台与 ISO 14443A 兼容，其帧格式采用 NFC-A 的标准帧。数据封装格式如图 4-59 所示。

数据(命令或响应)				[EOD]	
Byte 1	Byte 2	...	Byte *n*	[CRC_A1]	[CRC_A2]

图 4-59　类型 4A 标签平台命令响应封装格式

4.8.2 命令

类型 4A 标签平台用于支持类型 ISO 14443A 标签的读写器或卡模拟设备，也就是说，配置为标签 4A 平台的轮询设备作为 ISO 14443A 标签的读写器，配置为类型 4A 标签平台的侦听设备模拟成 ISO 14443A 标签。类型 4A 标签平台的设备激活命令及响应如表 4-8 所示。

表 4-8 类型 4A 标签平台设备激活命令及响应

命令	响应
RATS	ATS

轮询设备发送 RATS (Request for Answer to Select) 命令，与侦听设备协商最大的帧长度和比特传输速率，其消息格式如图 4-60 所示。

其中，PARAM 参数的编码格式如图 4-61 所示。

Byte 1	Byte 2
E0h	PARAM

图 4-60 RATS 命令格式

b8	b7	b6	b5	b4	b3	b2	b1	描述
×	×	×	×					FSDI
				×	×	×	×	DID

图 4-61 PARAM 参数编码格式

其中，FSDI (Frame Size for proximity coupling Device Integer) 用于指示 FSD (Frame Size for proximity coupling Device)。FSD 是表示轮询设备能够接收的最大帧长度，如图 4-62 所示。

FSDI	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h～Eh	Fh
FSD(bytes)	16	24	32	40	48	64	96	128	256	预留	256

图 4-62 FSD 的取值

DID (Device Identification Number) 用于在设备激活过程中对侦听设备寻址。如果目标侦听设备不支持 DID，或者不存在同时激活多个侦听设备的情况，则不使用 DID。

侦听设备收到 RATS 后，发送 ATS (Answer To Select) 消息进行响应。ATS 的消息格式如图 4-63 所示。

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6～6+k-1
TL	T0	TA(1)	TB(1)	TC(1)	T1 … Tk

图 4-63 ATS 消息格式

TL 称为长度字节，用于表示 ATS 消息的长度，包括 TL 字节。在 Digtial 规范中，规定 TL 表示的长度不能超过 20 字节。

T0 的编码格式如图 4-64 所示。b4～b1 用于表示侦听设备所支持的最大帧长 FSCI。b7～b5 分别用于表示 ATS 消息中是否包含 TC (1)、TB (1) 和 TA (1)。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0								预留
	×							设置为1，表示包含TC(1)
		×						设置为1，表示包含TB(1)
			×					设置为1，表示包含TA(1)
				×	×	×	×	FSCI

图 4-64 T0 编码格式

TA (1) 定义了侦听设备支持的比特传输速率。TB (1) 包含 FWI (Frame Waiting Time) 和 SFGI。其中，SFGI 在 4.3.2 节已经介绍过。FWT 定义了侦听设备收到轮询设备

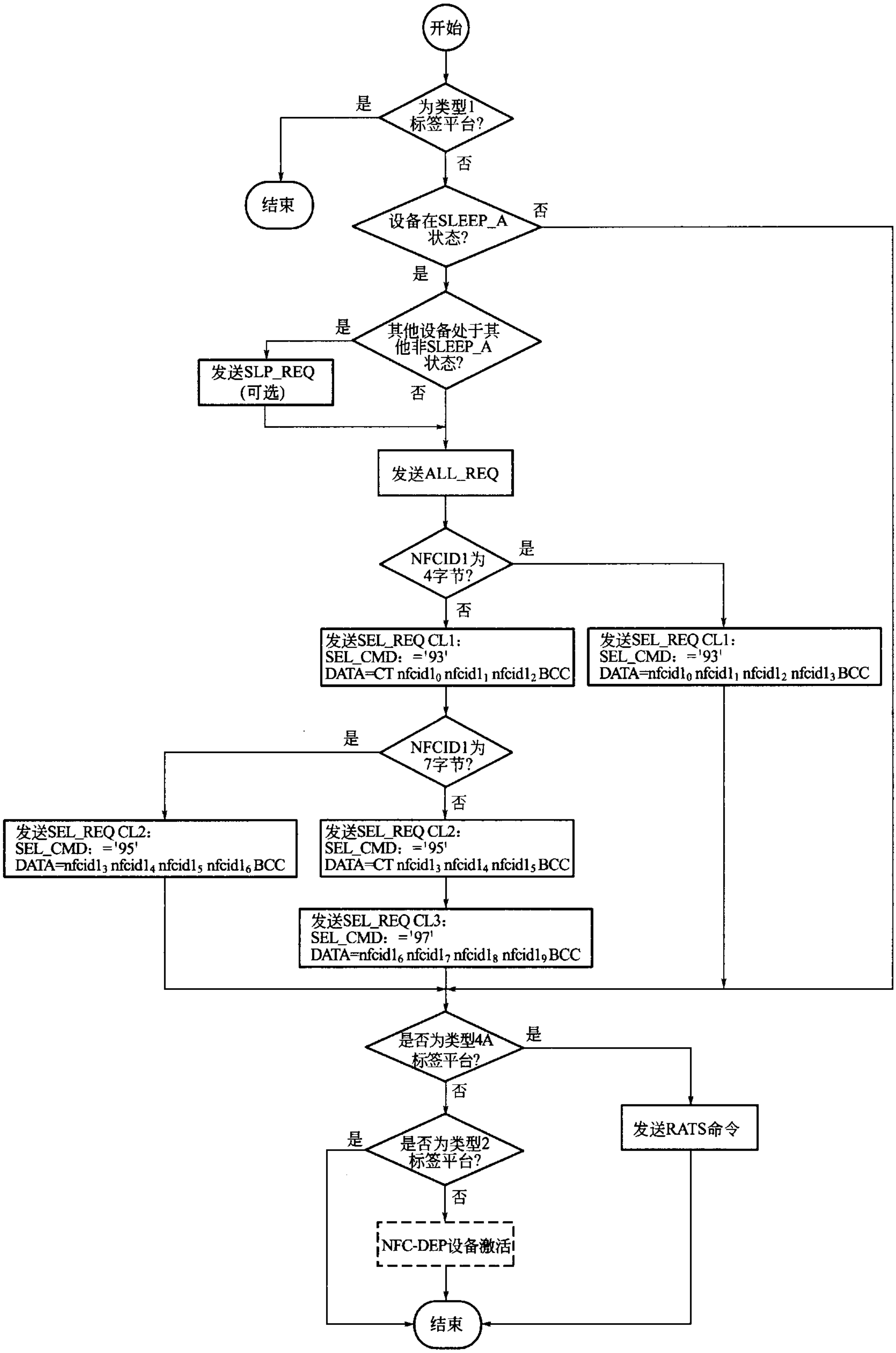


图 4-65 类型 4A 标签平台轮询设备的激活流程

的消息后，在回复之前的最大等待时间。TC (1) 描述了侦听设备是否支持高级协议特性、NAD 和 DID。在 Digital 1.0 规范中，不使用高级协议特性及 NAD。

T1 到 Tk 表示历史字节，是可选的。这里的历史字节是指在 ISO 14443 标准的演进过程中因为一些遗留问题所产生的字节。历史字节可以包含一些通用的信息，但是规范没有指定这些通用信息的内容，因此历史字节常被厂商用来携带一些私有信息。

类型 4A 标签平台在设备激活阶段使用了 RATS 命令和 ATS 响应，在数据读写过程中使用了 ISO-DEP 协议。关于数据读写的命令，将在 4.11 节介绍。

4.8.3 设备激活

类型 4A 标签平台与 NFC-A 的冲突检测过程一致。在设备激活过程中，类型 4A 标签平台采用 RATS/ATS 激活设备。在类型 4A 标签平台的设备激活过程中，配有类型 4A 标签平台的读写器和类型 4A 标签/卡模拟设备进行参数协商。协商完成后，二者交互数据，如图 4-65 所示。其中，步骤 1～步骤 5 与图 4-56 中所示相同。当轮询设备判断要激活的侦听设备配置为类型 4A 标签平台后，发送 RATS 命令，并接收 ATS 响应协商相关参数，完成设备激活的流程。

图 4-66 描述了侦听设备激活的状态转移图。当侦听设备收到 RATS 命令时，进入 CARD EMULATOR 4A 状态。在该状态下，轮询设备和侦听设备可以交换数据。

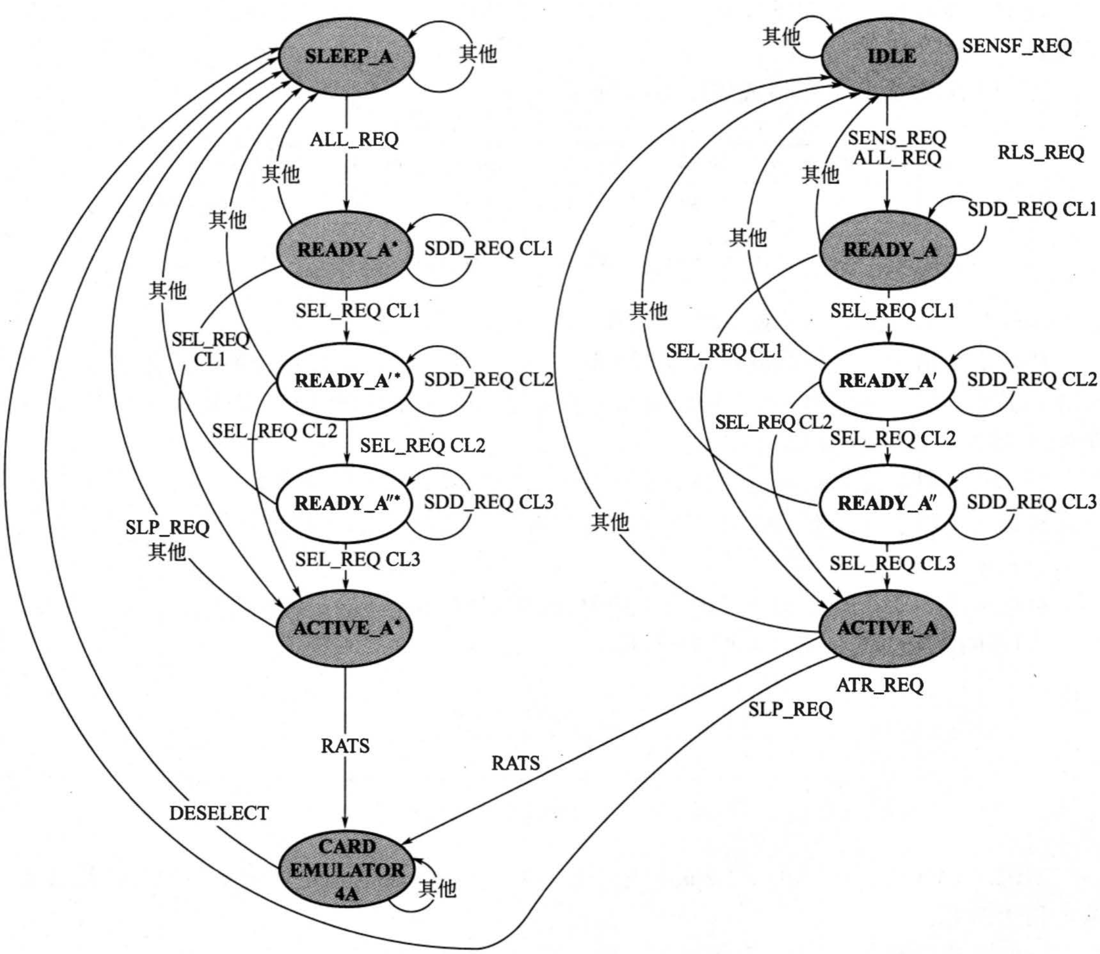


图 4-66 类型 4A 标签平台侦听模式下设备激活状态转移图

4.9 类型 4B 标签平台

4.9.1 封装格式

类型 4B 标签平台与 ISO 14443B 兼容，其帧格式和数据封装格式采用 NFC-B 的帧格式和数据封装格式。

4.9.2 命令

类型 4B 标签平台用于支持类型 ISO 14443B 标签的读写器或卡模拟设备，也就是说，配置为标签 4B 平台的轮询设备作为 ISO 14443B 标签的读写器，配置为类型 4B 标签平台的侦听设备模拟成 ISO 14443B 标签。类型 4B 标签平台的设备激活命令及响应如表 4-9 所示。

表 4-9 类型 4B 标签平台设备激活命令及响应

命 令	响 应
ATTRIB Command	ATTRIB Response

轮询设备和侦听设备通过交互 ATTRIB 命令和 ATTRIB 响应来协商参数，完成设备激活过程。

ATTRIB 命令的消息格式如图 4-67 所示。

Byte 1	Byte 2～5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10～10+*k*–1
1Dh	NFCID0	Param 1	Param 2	Param 3	Param 4	Higher layer-INF

图 4-67 ATTRIB 命令消息格式

NFCID0 为要激活侦听设备的 NFCID0。

Param 1 用于描述轮询设备的最小 TR0 和 TR1 要求，以及数据交换时是否包含 SOS 和 EOS。其中，TR0 和 TR1 用于轮询帧和轮询帧之间的间隔保证时间，即轮询帧的 EOS 和侦听帧 SOS 之间的保证时间。

Param 2 用于描述轮询设备和侦听设备间的比特传输速率以及 FSDI。

Param 3 用于协议类型的确认，如询问侦听设备是否与 ISO 14443 兼容。

Param 4 用于描述 DID。

Higher layer-INF 指上层命令在 ISO-DEP 协议的 INF 域中传输。INF 域将在 4.11 节介绍。

ATTRIB 响应消息格式如图 4-68 所示。

Byte 1		Byte 2～2+*n*–1
MBLI	DID	Higher layer-Response

图 4-68 ATTRIB 响应消息格式

MBLI（Maximum Buffer Length Index）用于侦听设备通知轮询设备关于自己能够接收帧的最大缓存。

DID 的值要与 ATTRIB 命令中的一致。

Higher layer-Response 用于对 ATTRIB 命令中 Higher layer-INF 的响应。

4.10 半双工通信

NFC 设备间采用半双工通信，即任何时候只能有一个设备发送数据。在 Digital 规范中，设备之间有主从关系，即处于轮询模式的设备称为主设备，处于侦听模式的设备称为从设备。轮询设备发送轮询帧，并等待侦听帧的到来；侦听设备在发送侦听帧之前需要等待轮询帧的到来。

半双工通信协议实际上描述了 NFC 设备之间通信共有的机制。对于具体的细节，每一个传输协议都不一样，如 NFC-DEP、ISO-DEP 都有自己的定义。类型 1 标签平台、类型 2 标签平台和类型 3 标签平台的通信也采用半双工机制，如图 4-1 所示。但是要强调的是，对于这三种标签平台来说，半双工通信协议只是提供了一种通信机制，至于数据交换和设备的去激活使用的命令和响应，不是由半双工通信协议来提供，而是由各自的标签平台和标签操作规范定义。

4.11 ISO-DEP

4.11.1 封装格式

ISO-DEP（ISO 14443 Data Exchange Protocol）即 ISO 14443-4 中定义的传输协议，用于类型 4A 和 4B 标签平台的数据交换和设备去激活。当类型 4A 和 4B 标签平台完成设备激活后，数据交换通过 ISO-DEP 协议传输。

ISO-DEP 的数据封装格式采用数据块（Block）的方式，如图 4-69 所示。SOD 由 PCB（Protocol Control Byte）和 DID 组成，DID 是可选的。EOD 由 2 字节校验码组成。对于类型 4A 标签平台，采用 CRC _ A；对于类型 4B 标签平台，采用 CRC _ B。

块(数据)				
SOD		负荷	EOD	
PCB	[DID]	[INF]	CRC_1	CRC_2

图 4-69 ISO-DEP 数据封装格式

PCB 携带了数据传输需要的一些控制信息，如数据块分成了三种类型：I-Block、R-Block 和 S-Block。其中，I-Block 用于承载应用层的数据，R-Block 用于确认（ACK 和 NAK），R-Block 没有 INF 域。S-Block 用于在读写器和卡模拟设备之间交换一些控制信息。有两种类型的 S-Block：WTX（Waiting Time eXtension）和 DESELECT。三种数据块的 PCB 编码格式如图 4-70～图 4-72 所示。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0							I-block
		×						预留
			×					如果设置为1，表示为信息数据块链
				×				如果设置为1，表示允许使用DID
					0			设置为0，表示不允许使用NAD
						1		设置为1
							×	数据块编号

图 4-70 I-Block 的 PCB

b8	b7	b6	b5	b4	b3	b2	b1	描述
1	0							R-block
		1						必须设置为1b
			×					设置为0，表示ACK 设置为1，表示NAK
				×				如果设置为1，表示允许使用DID
					0			设置为0
						1		预留
							×	数据块编号

图 4-71 R-Block 的 PCB

b8	b7	b6	b5	b4	b3	b2	b1	描述
1	1							S-block
		×	×					设置为00，表示DESELECT 设置为11，表示WTX
				×				如果设置为1，表示允许使用DID
					0			设置为0
						1		预留
							0	预留

图 4-72 S-Block 的 PCB

4.11.2 数据交换

I-Block 用于读写器和卡模拟设备之间的数据交换，应用数据负载包含在 INF 域中。与类型 1/2/3 标签不同，类型 4A/4B（ISO 14443 智能卡）采用 ISO 7816 所定义的文件系统，因此在数据的访问上采用 ISO 7816-4 的命令集来访问文件。I-Block 中的 INF 域主要用于承载 ISO 7816-4 的 APDU 指令。此外，INF 域可以承载 S-Block 中的非应用数据和状态信息。

I-Block 的交换需要通过 R-Block 确认，即 R-Block 的 ACK 和 NAK。I-Block 可以由读写器发送给卡模拟设备，也可以由卡模拟设备发送给读写器，因此 ACK 和 NAK 可以由读写器发送，也可以由卡模拟设备发送。发送 NAK 的情形如 I-Block 数据 CRC 出错，或者 I-Block编号出错等等。

当读写器和卡模拟设备交换的信息超过了 FSC 的定义，将被分成几个数据块，用数据块链标识位来表示。

S-Block 中的 WTX 用于当侦听设备（卡模拟设备或标签）需要更多 FWT 时间处理接收的数据块时，侦听设备发送 WTX 消息请求更多的时间。WTX 中包含 1 字节的 INF 域，用于表示请求的扩展时间。当轮询设备收到 WTX 请求消息后，返回 WTX 响应消息，该响应消息也包含 1 字节的 INF 域。如果 WTX 响应消息 INF 域中携带的扩展时间值与请求消息中的一致，表示轮询设备接收了请求。

4.11.3 设备去激活

S-Block 的 DESELECT 用于设备的去激活。当读写器发送 DESELECT 请求消息给卡模拟设备后，卡模拟设备应当返回 DESELECT 响应消息来完成设备去激活。

4.12 NFC-DEP

4.12.1 发起方和目标方

NFC-DEP 即 ISO 18092 中定义的数据传输协议，主要用于发起方（Initiator）和目标方（Target）之间的数据通信协议。在 ISO 18092 中，将通信的双方称为发起方和目标方，在 NFC Forum 的 Digital 规范中沿用 ISO 18092 的称谓，因此本节中以发起方和目标方来描述通信双方的角色。

发起方是指在通信过程中产生射频场并且发起 NFCIP-1 通信的设备。目标方是指采用负载调制或自己产生射频场来响应发起方的命令的设备。由于 ISO 18092 主要定义了点对点通信模式，因此在通信双方的角色称谓上与 ISO 14443 有所不同。ISO 14443 中是读写器和卡模拟设备/标签之间的通信，读写器只能处于轮询模式，因此也称为轮询设备；而卡模拟设备/标签一般处于侦听模式下，因此也称为侦听设备。在 ISO 18092 中，发起方和目标方可以处于轮询模式下，也可以处于侦听模式下。

4.12.2 封装格式

NFC-DEP 采用 NFC-A 的标准帧格式或 NFC-F 的帧格式，主要取决于 NFC 设备的配置。图 4-73 描述了 NFC-DEP 的数据封装格式。

数据							
SOD		负载				EOD	
SB(只针对NFC-A)	LEN	Byte 1	Byte 2	…	Byte n	CRC_1	CRC_2

图 4-73 NFC-DEP 数据封装格式

SOD 包含一个开始字节（Start Byte）和一个长度字节。开始字节只用于 NFC-A 技术，长度字节用于指示负载的长度。数据负载包括命令和响应，下面将详细介绍。EOD 包括 2 字节校验码。如果配置为 NFC-A 技术，使用 CRC _ A；如果配置为 NFC-F，使用 CRC _ F。

4.12.3 命令集

NFC-DEP 的命令集如表 4-10 所示。

表 4-10 NFC-DEP 命令集

命令	响应
ATR_REQ	ATR_RES
PSL_REQ	PSL_RES
DEP_REQ	DEP_RES
DSL_REQ	DSL_RES
RLS_REQ	RLS_RES

4.12.3.1 ATR_ REQ/ATR_ RES

ATR _ REQ 命令用于发起方对目标方进行设备激活，目标方向发起方发送 ATR _ RES 响应。ATR _ REQ 命令的编码格式如图 4-74 所示。

Byte 1	Byte 2	Byte 3～12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17～17+n
D4h	00h	**$NFCID3_I$**	**DID_I**	**BS_I**	**BR_I**	**PP_I**	[G_I0 ... G_In]

图 4-74 ATR _ REQ 命令编码格式

ATR _ REQ 中的字节 3～12 为 $NFCID3_I$，用于表示发起方的 NFCID。由于发起方可能支持 NFC-A 技术或者 NFC-F 技术，或者都支持，因此用 10 字节表示 NFCID。

发起方设备使用 DIDi 来标识多个被同时激活的设备。如果不允许同时激活多个设备，该值设置为 0。在 Digital 1.0 规范中不支持同时激活多个设备，因此该值被设置为 0。

BS_I 和 BR_I 表示主动通信中发起方支持的收发传输速率。由于在 Digital 1.0 规范中不支持主动通信，因此 BS_I 和 BR_I 均设置为 0。

PP_I（Presence of Optional Parameters）用于表示 LR_I（Length Reduction field）和是否包含可选参数。PP_I 的编码格式如图 4-75 所示。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0							预留
		×	×					**LR_I**
				0	0			预留
						×		如果包含通用字节, 设置为1
							×	如果支持NAD, 设置为1

图 4-75 PP_I 编码格式

b1 设置为 0，表示 Digital 规范中不使用 NAD。b2 用于表示 ATR _ REQ 命令是否包含通用字节 $G_I0 \cdots G_In$。b6 和 b5 表示 LR_I，用于限制目标方响应消息的大小，如图 4-76 所示。

b6	b5	描述
0	0	最大负荷为64字节
0	1	最大负荷为128字节
1	0	最大负荷为192字节
1	1	最大负荷为256字节

图 4-76 LR_I 编码格式

通用字节 G_I 是可选的，一般用于提供通用信息。通用字节也称为历史字节，一些对于协议的扩展可以通过通用字节来完成。

ATR _ RES 响应消息格式如图 4-77 所示。

Byte 1	Byte 2	Byte 3～12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18～18+n
D5h	01h	**$NFCID3_T$**	**DID_T**	**BS_T**	**BR_T**	TO	**PP_T**	[G_T0 ... G_Tn]

图 4-77 ATR _ RES 响应消息格式

$NFCID3_T$ 用于标识目标方的 NFCID 值，同样也是 10 字节，用于支持 NFCID1 和 NF-CID2。

DID_T 与 ATR _ REQ 命令中的 DID_I 值相同。

BS_T 和 BR_T 表示主动通信模式下，目标方支持的收发速率，均设置为 0。

TO 的 b4～b1 为 WT（Waiting Time，等待时间），用于目标方的响应等待时间。

PP_T 用于表示 LR_T 和是否包含可选参数，其编码格式与 PP_I 相同。LR_T 用于限制目标方的命令长度，其编码格式与 LR_I 相同。

通用字节 G_T 与 G_I 的概念相同。

4.12.3.2 PSL_ REQ/PSL_ RES

PSL _ REQ (Parameter Selection Request) 用于通信参数的切换。在设备激活之后，NFC-DEP 还能为接下来的数据通信重新协商通信参数，主要包括传输速率、最大负载长度。PSL _ REQ 命令是可选的，如果不发送 PSL _ REQ，就采用 ATR _ REQ/ATR _ RES 中协商的参数。PSL _ REQ 命令的消息格式如图 4-78 所示。

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
D4h	04h	DID	BRS	FSL

图 4-78 PSL _ REQ 命令格式

DID 与 ATR _ REQ 中的 DID 一致。

BRS 用于表示发起方和目标方选用的比特速率，其编码格式如图 4-79 所示。其中，DSI 表示从发起方到目标方的传输速率，DRI 表示目标方到发起方的传输速率。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0							预留
		×	×	×				DSI
					×	×	×	DRI

图 4-79 BRS 编码格式

Byte 1	Byte 2	Byte 3
D5h	05h	DID

图 4-80 PSL _ RES 消息格式

FSL 表示最大的命令和响应的长度（即负载的字节数），支持 64 字节，128 字节、192 字节和 254 字节。

PSL _ RES 的消息格式如图 4-80 所示。其中，DID 与 PSL _ REQ 中的 DID 一致。PSL _ RES 消息用于目标方确认发起方对通信参数的切换请求。

4.12.3.3 DEP_ REQ/DEP_ RES

DEP _ REQ (Data Exchange Protocol Request) 用于发起方与目标方交换数据，目标方向发起方响应 DEP _ RES 消息。DEP _ REQ 命令消息格式如图 4-81 所示。

数据交换协议头					数据字节		
Byte 1	Byte 2	Byte 3	Byte 4 (可选)	Byte 5 (可选)	Byte 6～6+k-1 (可选)		
D4h	06h	PFB	[DID]	[NAD]	[Data Byte 1]	…	[Data Byte k]

图 4-81 DEP _ REQ 命令消息格式

PFB (Protocol Format Byte) 用于携带数据传输相关的控制信息，如 PDU 的分类等。PDU (Protocol Data Units) 分为三类：

① 信息 PDU，用于携带应用层数据。

② ACK/NACK PDU，用于肯定或否定的确认。

③ 管理 PDU (Supervisory PDU)，用于在发起方和目标方之间交换控制信息。有两种类型的管理 PDU，即超时扩展 PDU 和 ATN PDU (Attention)。

信息 PDU 的 PFB 编码格式如图 4-82 所示。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0	0						信息PDU
			×					MI, 设置为1, 表示数据链
				×				NAD, 如果设置为1, 表示支持
					×			DID, 如果设置为1, 表示支持
						×	×	PNI

图 4-82 信息 PDU 的 PFB 编码格式

MI（More Information）用于表示数据链，即应用数据被分割成几个 PDU，这些 PDU 组成了一个数据链。

PNI（Packet Number Information）用于发起方向目标方发送数据的计数。

ACK/NACK PDU 的 PFB 编码格式如图 4-83 所示。

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	1	0						ACK/NACK PDU
			×					0：ACK 1：NACK
				×				NAD, 如果设置为1, 表示支持
					×			DID, 如果设置为1, 表示支持
						×	×	PNI

图 4-83 ACK/NACK PDU 的 PFB 编码格式

管理 PDU 的 PFB 编码格式如图 4-84 所示。当目标方需要更多的时间来处理接收的 PDU 时，它发送 RTOX（Response Timeout Extension），请求给发起方给予更多的处理时间。RTOX 请求包含在管理 PDU 中。

b8	b7	b6	b5	b4	b3	b2	b1	描述
1	0	0						管理PDU
			×					0：ATN 1：超时
				×				NAD, 如果设置为1, 表示支持
					×			DID, 如果设置为1, 表示支持
						0	0	预留

图 4-84 管理 PDU 的 PFB 编码格式

DEP _ RES 响应消息的格式如图 4-85 所示。

数据交换协议头					数据字节		
Byte 1	Byte 2	Byte 3	Byte 4 (可选)	Byte 5 (可选)	Byte 6～6+*k*−1	(可选)	
D5h	07h	PFB	[DID]	[NAD]	[Data Byte 1]	…	[Data Byte *k*]

图 4-85 DEP _ RES 响应消息格式

4.12.3.4 DSL_ REQ/DSL_ RES

DSL _ REQ（Deselect Request）用于发起方去激活目标方，目标方向发起方发送 DSL _ RES 进行响应。DSL _ REQ 命令的消息格式如图 4-86 所示。

DSL _ RES 的消息格式如图 4-87 所示。

Byte 1	Byte 2	Byte 3（可选）
D4h	08h	[DID]

图 4-86 DSL _ REQ 命令格式

Byte 1	Byte 2	Byte 3（可选）
D5h	09h	[DID]

图 4-87 DSL _ RES 消息格式

4.12.3.5 RLS_ REQ/RLS_ RES

发起方发送 RLS _ REQ（Release Request）来释放/去激活目标方，目标方向发起方响应 RLS _ RES 消息。DLS _ REQ 命令和 RLS _ REQ 命令的区别在于，RLS _ REQ 命令是让目标方返回 IDLE 状态，而 DLS _ REQ 命令是让目标方返回睡眠状态。RLS _ REQ 命令的消息格式如图 4-88 所示。

RLS _ RES 响应消息格式如图 4-89 所示。

Byte 1	Byte 2	Byte 3（可选）
D4h	0Ah	[DID]

图 4-88 RLS _ REQ 命令格式

Byte 1	Byte 2	Byte 3（可选）
D5h	0Bh	[DID]

图 4-89 RLS _ RES 响应消息格式

4.12.4 设备激活

配置为 NFC-DEP 协议的发起方和目标方通过 ATR _ REQ/ATR _ RES 激活设备。NFC-DEP 用于 NFC-A 和 NFC-F 两种技术，因此下面分别介绍这两种技术的设备激活过程。

基于 NFC-A 的 NFC-DEP 发起方设备激活流程如图 4-90 所示。其中，设备唤醒的过程与图 4-56 所示步骤 1～步骤 5 一致。NFC-DEP 的设备激活流程如下所述：

步骤 1：发起方判断目标方配置为 NFC-DEP 协议。

步骤 2：发起方发送 ATR _ REQ 命令，与目标方进行参数协商。

步骤 3：发起方确定是否需要更新通信参数。如果是，执行步骤 4，否则执行结束。

步骤 4：发起方发送 PSL _ REQ 命令，更新通信参数，执行结束。

基于 NFC-F 的 NFC-DEP 发起方设备激活流程如图 4-91 所示，具体步骤如下所述：

步骤 1：发起方判断待激活的目标方是否支持类型 3 标签平台。如果支持，执行结束，否则执行步骤 2。对于类型 3 标签平台来说，没有别的设备激活步骤，在冲突检测完成后就可以收发数据。

步骤 2：发起方判断目标方是否处于休眠状态（SLEEP _ AF）。如果是，发送 SENSF _ REQ 命令将其唤醒后，执行步骤 3，否则直接执行步骤 3。

步骤 3：发起方向目标方发送 ATR _ REQ 命令激活设备。

步骤 4：发起方判断是否需要更新通信参数，如果是，发送 PSL _ REQ 后执行结束，否则直接执行结束。

图 4-92 所示为 NFC-DEP 目标方的状态转移图。

对于 NFC-A 技术来说，在冲突检测完毕后，目标方将进入 ACTIVE _ A 或 ACTIVE _ A* 状态。在该状态下，如果收到 RATS 命令，进入 CARD EMULATOR 4A 状态；如果收到 ATR _ REQ 命令，将进入 ATR _ READY _ A 状态，该状态说明目标方已经做好了接收数据的准备，一旦收到 DEP _ REQ 命令就进入数据收发状态 TARGET _ A。另外，在

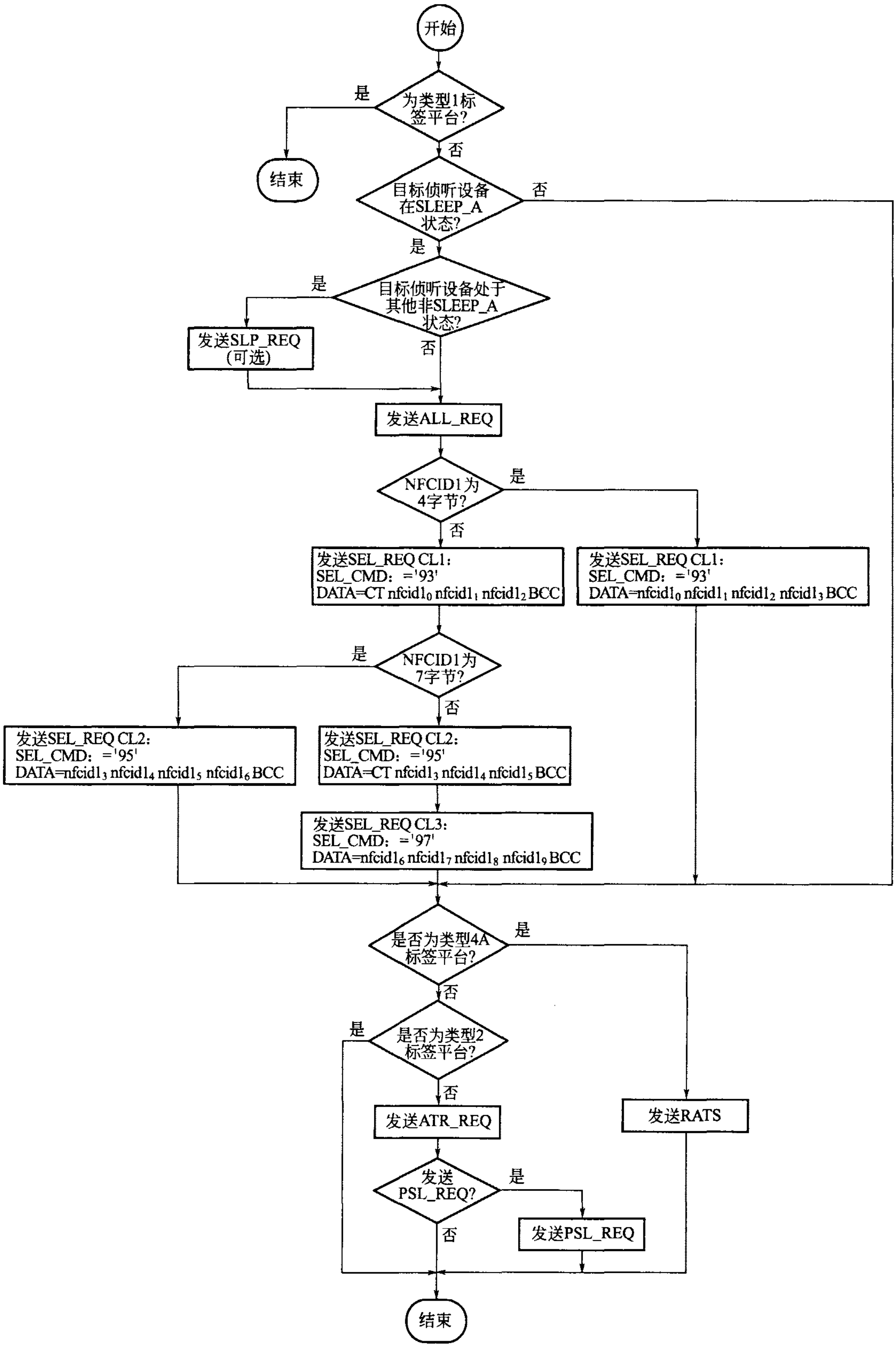


图 4-90 基于 NFC-A 的 NFC-DEP 发起方设备激活流程

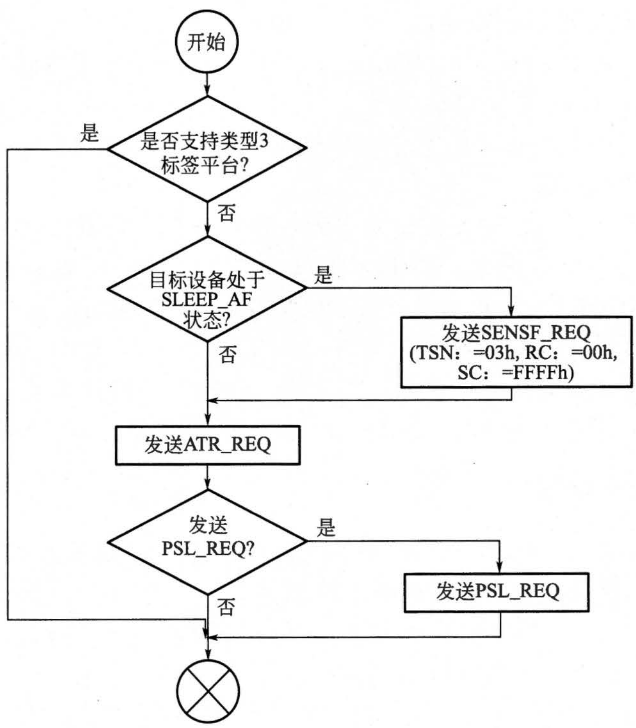


图 4-91 基于 NFC-F 的 NFC-DEP 发起方设备激活流程

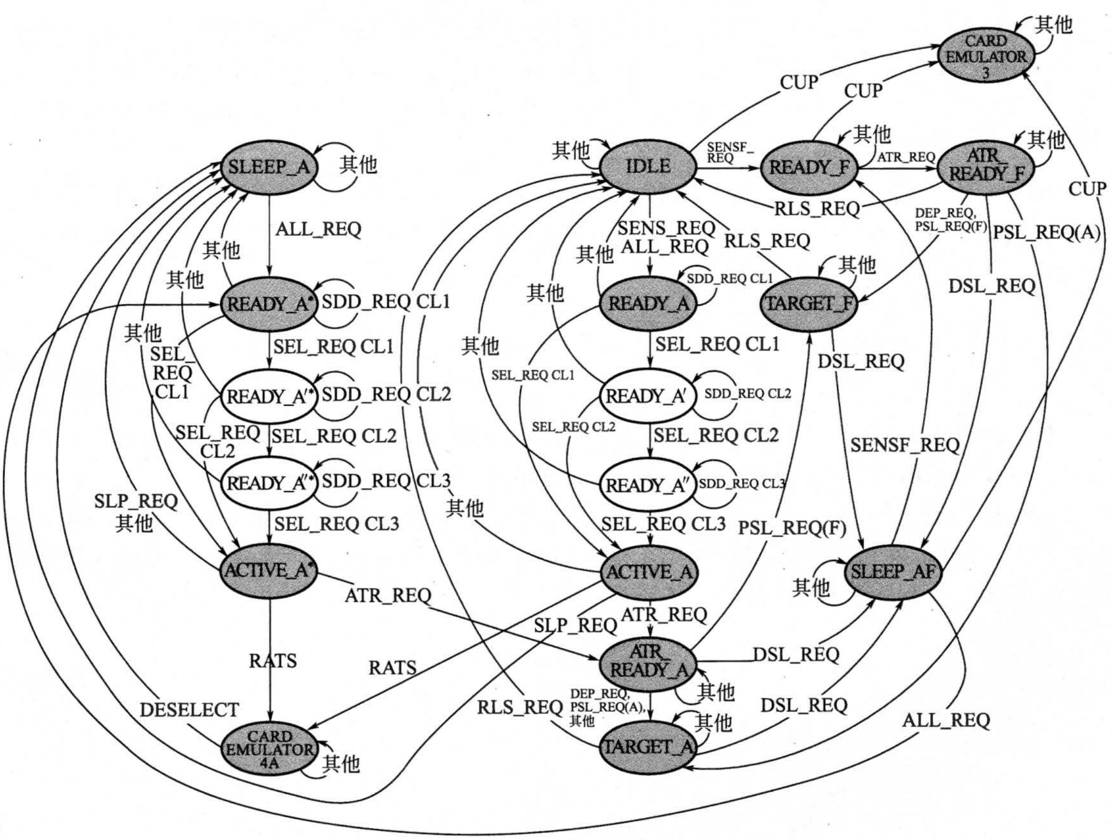


图 4-92 NFC-DEP 目标方设备激活状态转移图

ATR _ READY _ A 状态下也可以接收 PSL _ REQ 来更新通信参数。这里值得注意的是，PSL _ REQ 命令还可以将 NFC-A 技术切换为 NFC-F 技术，也可以将 NFC-F 技术切换成 NFC-A 技术，前提是发起方和目标方同时支持 NFC-A 和 NFC-F 技术。当在 ATR _ READY _ A 状态下收到切换技术的 PSL _ REQ 命令后，目标方将跳转到 TARGET _ F 状态，接收 NFC-F 的数据信息。

对于 NFC-F 技术来说，在冲突检测完毕后，目标方进入到 READY _ F 状态。在该状态下，当收到 CUP 命令时，进入 CARD EMULATOR 3 状态；如果收到 ATR _ REQ 命令，将进入 ATR _ READY _ F 状态，该状态说明目标方已经做好了接收数据的准备，一旦收到 DEP _ REQ 命令就进入数据收发状态 TARGET _ A。在 ATR _ READY _ F 状态下，也可以接收 PSL _ REQ 来更新通信参数。同样地，当收到切换技术的 PSL _ REQ 命令后，目标方将跳转到 TARGET _ A 状态，接收 NFC-A 的数据信息。

4.12.5 设备去激活

发起方通过发送 DSL _ REQ 命令或 RLS _ REQ 命令来去激活设备。对于发起方来说，设备去激活流程如图 4-93 所示。

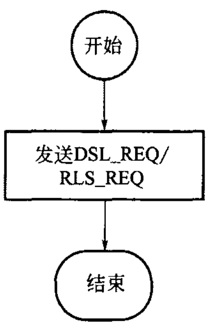


图 4-93 设备去激活

目标方的设备去激活状态转移如图 4-92 所示。

对于 NFC-A 技术来说，目标方在 ATR _ READY _ A 状态下收到 DSL _ REQ 命令，将进入 SLEEP _ AF 状态。在 TARGET _ A 状态下收到 DSL _ REQ 命令，也会进入 SLEEP _ AF 状态；收到 RLS _ REQ 命令，返回 IDLE 状态。

对于 NFC-F 技术来说，目标方在 ATR _ READY _ F 状态下收到 DSL _ REQ 命令，将进入 SLEEP _ AF 状态。在 TARGET _ F 状态下收到 DSL _ REQ 命令，也会进入 SLEEP _ AF 状态；收到 RLS _ REQ 命令，返回 IDLE 状态。

可以看出，RLS _ REQ 命令会使目标方返回 IDLE 状态，而 DLS _ REQ 命令让目标方返回 SLEEP _ AF 状态。需要注意的是，NFC-A 和 NFC-F 技术可以相互切换，两种技术可共用同一个休眠状态。

4.13 小结

本章介绍了 Digital 规范和 Activity 规范，以及两个规范之间的关系；分别介绍了 NFC-A、NFC-B 和 NFC-F 三种技术以及三种技术的子集：类型 1 标签平台、类型 2 标签平台、类型 3 标签平台、类型 4A 标签平台和类型 4B 标签平台；最后详细介绍了 ISO-DEP 和 NFC-DEP 两种传输协议。本章在介绍 Digital 规范命令集的同时，穿插介绍 Activity 规范中如何使用这些命令集来完成 NFC 通信流程。

Retro Workshop 果粉工作室

第5章 LLCP规范

5.1 LLCP概述

5.1.1 LLCP协议架构

LLCP[1]（Logical Link Control Protocol，逻辑链路控制协议）规范描述了 NFC 协议栈 LLC（逻辑链路控制）的功能和特性。LLCP 位于 Digital 协议之上，准确地说是 LLCP 基于 NFC-DEP 传输协议，LLCP 与 NFC-DEP 一起应用在点对点通信中，如图 5-1 所示。这里需要注意的是，为了保持与规范术语统一，在描述协议时用 LLCP 表示，描述 LLCP 的逻辑链路控制功能时用 LLC 表示。

对比 OSI 七层参考模型，LLCP 对应 OSI 的数据链路层上半部分功能，即 MAC（Medium Access and Control，媒质访问控制）层之上的 LLC 功能。LLCP 中定义了一套映射机制来绑定不同的传输协议，如 NFC-DEP 协议。与 IEEE 802.11（无线局域网）协议相比，LLCP 对应 IEEE 802.11 的 LLC。LLCP 与 OSI 参考模型及 IEEE 802.11 的比较如图 5-2 所示。

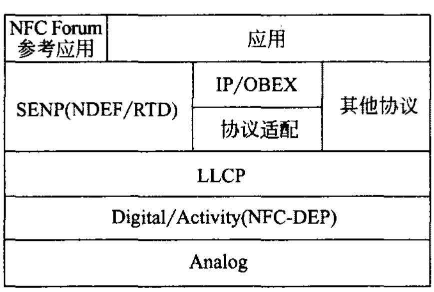


图 5-1　LLCP 在 NFC 协议栈中的位置

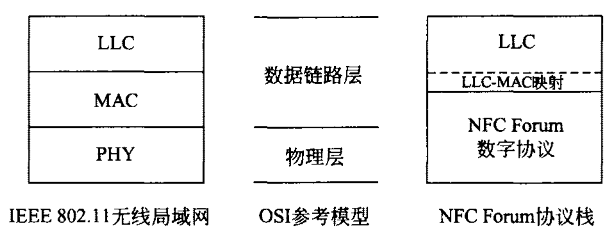


图 5-2　LLCP 与 OSI 参考模型/IEEE 802.11

LLCP 为上层协议提供数据传输服务，包括面向连接的服务和无连接服务，也包括链路的激活/去激活和维护功能。

LLCP 为了支持点对点通信，采用异步平衡通信模式（Asynchronous Balanced Mode,

---

[1] NFC Forum. Logical Link Control Protocol Technical Specification 1.1.

ABM)。在 MAC 层即 Digital 规范中，NFC 设备之间是主从关系，即发起方为主设备，目标方为从设备。对于点对点通信模式来说，通信双方是对等的，因此 LLCP 使用了 ABM 支持点对点通信。ABM 是一种高级数据链路控制的数据通信方式，其中异步是指两个通信设备在没有公共时钟的情况下传输数据；平衡是指在两个设备之间对等地进行点到点通信，消除了数据链路两端有主设备、从设备之分的“不平衡性”。

LLCP 还提供了协议复用功能，同时支持上层协议的不同实例。协议复用通过服务接入点（Service Access Point，SAP）来区分不同的上层协议或应用。

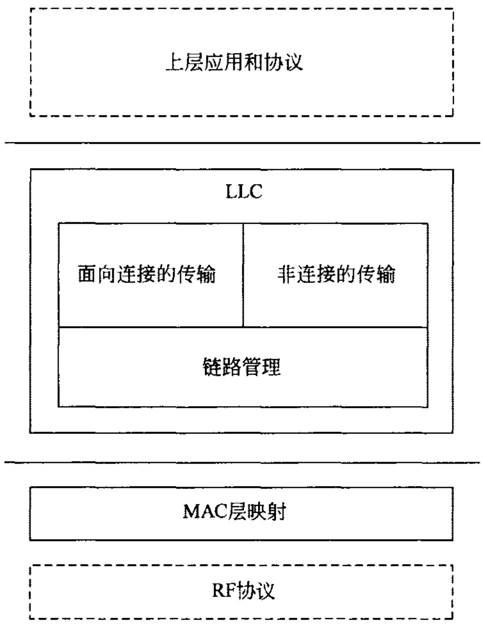


图 5-3 LLCP 协议架构

LLCP 的协议架构如图 5-3 所示。

MAC 层映射将射频通信协议（如 NFC-DEP）集成到 LLCP 架构中。LLCP 规范提及的 MAC 层映射，就是将 LLCP 映射到 NFC-DEP 协议。这里映射的具体含义是 LLC 和 NFC-DEP 之间数据的封装/解封装，即能够将 LLCP 定义的 LLC PDU（Protocol Data Unit，协议数据单元）作为数据负载适配进 NFC-DEP 的 DEP _ REQ/RES 消息，并且将收到的 DEP _ REQ/RES 消息中的数据负载映射为 LLC PDU。

链路管理功能主要负责所有面向连接或无连接 LLC PDU 传输的有序化，以及 PDU 的聚合（Aggregation）和去聚合（Disaggregation），同时保证了异步平衡模式通信并提供链路状态管理。

面向连接的传输提供了连接的建立和释放，以及数据交换。无连接传输服务提供了无需确认的数据交换。面向连接传输服务和无连接传输服务为上层应用或协议提供两种类型的传输服务。

5.1.2 LLCP 链路服务类型

LLCP 提供了两种传输服务：面向连接的传输服务和无连接传输服务。根据这两种传输服务，LLCP 定义了三种链路服务类型：

① 链路服务类型 1：仅支持无连接传输服务的链路。

② 链路服务类型 2：仅支持面向连接传输服务的链路。

③ 链路服务类型 3：同时支持面向连接传输服务和无连接传输服务的链路。

5.1.3 无连接传输

无连接传输无需对传输的数据进行确认，一般用于上层应用或协议提供了必要的恢复功能和序列化服务的情况，即由上层应用或服务来保证数据传输的序列化。无连接传输无需事先建立或释放连接，数据的 SSAP（Source Service Access Point，源服务接入点）和 DSAP（Destination Service Access Point，目的服务接入点）可以唯一地标识一条逻辑数据链路。

5.1.4 面向连接的传输

面向连接的传输服务为数据传输提供有序的、可靠的服务。在数据传输之前，需要在

SSAP 和 DSAP 之间建立连接。连接建立好之后，就可以传输数据。传输的数据需要对方确认，通过对数据编号和滑动窗口来控制数据发送。数据连接由 SSAP 和 DSAP 组成的二元组唯一地标识。当建立连接时，DSAP 用于标识对方设备要接收数据的服务端口，而 SSAP 标识本地的服务端口。在通信结束时，需要释放连接。

5.1.5 SAP

SAP 称为服务接入点，上层应用或协议通过 SAP 获取 LLCP 提供的传输服务，因此 SAP 用于标识 LLCP 上层应用或协议的实例。SAP 分为 SSAP 和 DSAP。SSAP 用于标识本地设备的服务接入点，DSAP 用于标识远端设备的服务接入点。SSAP 和 DSAP 组成的二元组用于 LLC PDU 寻址。如图 5-4 所示，设备 A 的应用程序 1 通过 LLC 向设备 B 的应用程序 1 发送 LLC PDU，那么在 LLC PDU 中，SSAP 为设备 A 的 SAP1，而 DSAP 为设备 B 的 SAP1。同理，设备 B 的应用程序 1 向设备 A 的应用程序发送 LLC PDU，那么 LLC PDU 中的 SSAP 为设备 B 的 SAP1，而 DSAP 为设备 A 的 SAP1。

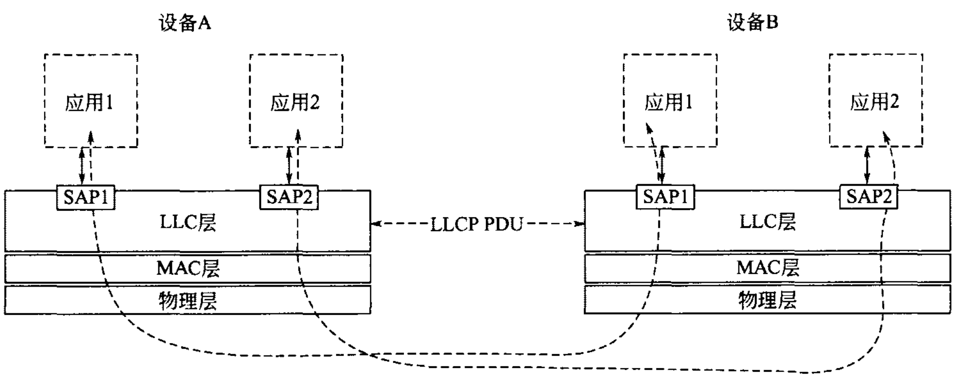


图 5-4 SAP 举例

SAP 的取值如表 5-1 所示。

表 5-1 SAP 取值

SSAP/DSAP	描　述
00h～0Fh	由 NFC Forum 所定义的 SAP 值
10h～1Fh	用于表示注册在本地的 SAP,这些 SAP 值可以通过 SDP 协议被远端 LLCP 获得
20h～3Fh	用于表示注册在本地的 SAP,但是这些 SAP 不能通过 SDP 被远端 LLCP 获得

已知类型 00h～0Fh 是指由 NFC Forum 定义的 SAP 值。SAP=0 表示 LLC 链路管理单元（如图 5-3 中的链路管理功能），链路管理命令如 SYMM、AGF，PAX 的 PDU 中的 SSAP/DSAP 都设置为 0。SAP=1 表示 SDP（Service Discovery Protocol，服务发现协议），用于本地 LLC 通过服务名称来获取远端设备 LLC 的 SAP 值。这里本地和远端是相对的概念，以图 5-4 举例，对于设备 A 来说，自己的 SAP1 和 SAP2 称为本地 SAP，而设备 B 的 SAP1 和 SAP2 称为远端设备的 SAP；反之，对于设备 B 来说，自己的 SAP1 和 SAP2 称为本地 SAP，而设备 A 的 SAP1 和 SAP2 称为远端设备的 SAP。

10h～1Fh 用于注册在本地的 SAP，这些 SAP 值可以通过 SDP 协议被远端 LLC 获得。也就是说，这些 SAP 值一般对应一个公开的服务名称，因此可以通过 SDP 协议来发现。

20h～3Fh 用于表示注册在本地的 SAP，但是这些 SAP 不会通过 SDP 被远端 LLC 获得，一般用于产品的内部实现。

5.2 LLCP 消息

5.2.1 LLC PDU 格式

LLC PDU 的格式如图 5-5 所示。

每一个 PDU 都包含两个地址域：SSAP 和 DSAP。DSAP 用于表示这个 PDU 应当被发给远端设备 LLC 的哪个 SAP。SSAP 表示这个 PDU 是由本地 LLC 的哪个 SAP 发出的。SSAP 和 DSAP 的取值在 5.1.5 节中已介绍。

LLCP头				LLCP负荷
DSAP	PTYPE	SSAP	Sequence(序列) N(S) | N(R)	信息
6位	4位	6位	0或8位	$M\times8$位

图 5-5　LLC PDU 格式

每个 PDU 都有一个 4bit 的类型域 PTYPE 来表示其类型，如表 5-2 所示。其中的 PDU 消息将在 5.2.2 节及 5.2.3 节具体介绍。

表 5-2　PDU 类型取值

PDU 类型	PTYPE	链路服务类型	PDU 类型	PTYPE	链路服务类型
SYMM	0000	1,2,3	FRMR	1000	2,3
PAX	0001	1,2,3	SNL	1001	1,2,3
AGF	0010	1,2,3	保留	1010	
UI	0011	1,3	保留	1011	
CONNECT	0100	2,3	I	1100	2,3
DISC	0101	1,2,3	RR	1101	2,3
CC	0110	2,3	RNR	1110	2,3
DM	0111	1,2,3	保留	1111	

序号（Sequence）用于表示 PDU 的序列号，包含两部分，N(S) 表示发送的序列号，N(R) 表示接收的序列号。在目前的 LLCP 规范中，只有 I PDU、RR PDU 和 RNR PDU 具有序列号，其他 PDU 都没有序列号。

信息域（Information）用于表示所携带的数据。信息域可以为空。

5.2.2 LLC PDU 类型

5.2.2.1 SYMM

SYMM（Symmetry）PDU 消息格式如图 5-6 所示。

当逻辑链路上没有需要发送的 PDU 时，LLC 通过发送 SYMM PDU 来维护链路。SYMM PDU 中的 SSAP 和 DSAP 均设置为 0，表示这个 PDU 是由链路管理单元发送和接收的，用于链路的管理功能。PTYPE 为 0，表示该 PDU 为 SYMM PDU。SYMM PDU 一般用于维护链路，如果没有收到其他 PDU，并且在规定时间内没有收到 SYMM PDU，远端设备 LLC 会认为逻辑链路已经断开。

DSAP	PTYPE	SSAP
000000	0000	000000

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1

图 5-6 SYMM PDU 格式

5.2.2.2 PAX

PAX（Parameter Exchange）PDU 格式如图 5-7 所示。PAX PDU 用于在 LLC 链路间交换参数。PAX 的 SSAP 和 DSAP 设置为 0，表示这个 PDU 是由链路管理单元发送和接收的，用于链路管理功能。在 PAX 的信息域中包含参数列表，参数可以是 LLC 协议版本号、LLC PDU 所能携带的最大信息数、已知的服务列表、链路的超时时间等，这些参数将会在后续章节介绍。

DSAP	PTYPE	SSAP	信息
00000000	0001	00000000	参数表

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	…	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n-1

图 5-7 PAX PDU 格式

5.2.2.3 AGF

AGF PDU 格式如图 5-8 所示。AGF 又称聚合帧（Aggregated Frame），用于将多个 LLC PDU 集合在一起通过一个 AGF PDU 发送给远端 LLC 链路管理单元。远端 LLC 链路管理单元将 AGF 信息域中的多个 LLC PDU 分解出来，根据 SSAP/DSAP 进行分发，如分发给上层应用或协议，或链路管理单元等。

DSAP	PTYPE	SSAP	信息
00000000	0010	00000000	LLC PDU

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	…	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n-1

图 5-8 AGF PDU 格式

AGF 信息域中可以包含多个 LLC PDU，每一个 PDU 的前面都需要加 2 字节长度域表示该 LLC PDU 的长度，如图 5-9 所示，这样，收到 AGF PDU 的远端设备的 LLC 链路管理单元才能根据长度将信息域中的每一个 PDU 解码出来。AGF PDU 要包含至少两个 LLC PDU，SYMM 不能通过 AGF 传送。另外，AGF 不能包含在 AGF 中。AGF PDU 的最大长度是由 MIU（Maximum Information Unit，最大信息单元）决定的。

5.2.2.4 UI

UI（Unnumbered Information）PDU 用于无连接传输服务的数据传输，其消息格式如图 5-10 所示，SSAP/DSAP 用于指定发送和接收 UI PDU 的 SAP，服务数据单元 SDU 表示

Length L			LLC PDU	
15 0		7 0	…	7 0
byte 0	byte 1	byte 2	…	byte L+1

图 5-9 AGF 信息域格式

上层协议或应用的数据。对于 PDU 和 SDU 的区别，在 OSI 参考模型中，PDU 用于对等实体之间的数据，SDU 用于上、下层之间的数据。以 LLCP 为例，简单来说，PDU 在本地 LLC 和远端设备 LLC 之间传输。LLC PDU 的格式如图 5-5 所示。LLCP 的上层应用或上层协议的数据称为服务数据单元。LLCP 是为上层应用或协议提供传输服务的，即上层应用或协议的数据通过 LLC PDU 发送到对端。

DSAP	PTYPE	SSAP	信息
DDDDDD	0011	SSSSSS	服务数据单元(SDU)

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	…	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n−1

图 5-10 UI PDU 格式

5.2.2.5 CONNECT

CONNECT 是用于面向连接传输服务中请求连接建立的 PDU，其格式如图 5-11 所示。SSAP/DSAP 用于表示需要在哪两个 SAP 之间建立连接，信息域中的参数列表包含建立连接需要协商的参数。

DSAP	PTYPE	SSAP	信息
DDDDDD	0100	SSSSSS	参数表

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	…	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n−1

图 5-11 CONNECT PDU 格式

5.2.2.6 DISC

DISC（Disconnect）PDU 用于面向连接传输服务中释放连接或者去激活 LLC 链路，消息格式如图 5-12 所示。

DSAP	PTYPE	SSAP
DDDDDD	0101	SSSSSS

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1

图 5-12 DISC PDU 格式

5.2.2.7 CC

如果远端设备 LLC 接受了本地设备 LLC 的连接建立请求，CC（Connection Complete）PDU 用于确认 CONNECT PDU，其消息格式如图 5-13 所示。如果远端设备 LLC 拒绝了本

地设备 LLC 的连接建立请求，将发送 DM PDU。

DSAP	PTYPE	SSAP	信息
DDDDDD	0110	SSSSSS	参数表

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	…	7	6	5	4	3	2	1	0
字节偏移0								字节偏移1								字节偏移2								…	字节偏移n−1							

图 5-13 CC PDU 格式

5.2.2.8 DM

DM（Disconnected Mode）PDU 用于报告关于 LLC 数据连接或链路断开的原因，其数据格式如图 5-14 所示。

DSAP	PTYPE	SSAP	信息
DDDDDD	0111	SSSSSS	原因

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
字节偏移0								字节偏移1								字节偏移2							

图 5-14 DM PDU 格式

其中，信息域用于表示连接或链路断开的原因，其具体含义如表 5-3 所示。

表 5-3 信息域的具体含义

原因代码	描　述
00h	用于收到 DISC 的确认
01h	收到面向连接传输的 PDU，但是 DSAP 不可用或未被激活
02h	用于确认 CONNECT PDU。收到 CC PDU，但是相关的参数不满足连接建立的条件
03h	远端设备的 LLCP 收到 CONNECT PDU，但是被上层应用或协议拒绝
10h	表示永久的不接受包含该 DSAP 的 CONNECT PDU
11h	表示永久不接受任何 CONNECT PDU
20h	表示暂时不接受包含该 DSAP 的 CONNECT PDU
21h	表示暂时不接受任何 CONNECT PDU
其他值	不能用于 DM

5.2.2.9 FRMR

当 LLC 收到一个不正确的 PDU 后，将发送 FRMR（Frame Reject）PDU 表示拒绝，其数据格式如图 5-15 所示。

如果 W 设置为 1，表示 LLC 收到了一个无效或格式不正确的 PDU。如果 I 设置为 1，表示收到的 PDU 信息域数据不正确。如果 R 设置为 1，表示接收到的 PDU 接收序列号不正确。如果 S 设置为 1，表示收到的 PDU 发送序列号不正确。

如果被拒绝的 PDU 中包含 Sequence 域，那么 FRMR 的 Sequence 域要与其一致。如果被拒绝的 PDU 没有 Sequence 域，那么 FRMR 的 Sequence 域设置为 0。

V(S)、V(R)、V(SA)、V(RA) 分别表示发送 FRMR 的 LLC 当前的发送、接收、发

DSAP	PTYPE	SSAP	信息						
DDDDDD	1000	SSSSSS	W I R S	PTYPE	Sequence(序列)	V(S)	V(R)	V(SA)	V(RA)

7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0
字节偏移0	字节偏移1	字节偏移2	字节偏移3	字节偏移4	字节偏移5

图 5-15　FRMR PDU 格式

送确认和接收确认的状态变量。

5.2.2.10　I

I（Information）PDU 用于面向连接传输服务中发送服务数据单元 SDU，其数据格式如图 5-16 所示。I PDU 包含 Sequence 域，用于对 I PDU 的收、发进行编号。N（S）指示该 I PDU 的编号；N(R) 用于确认已经收到的 I PDU，即表示 LLC 已经成功地收到编号为 N(R)－1 的 I PDU。简单来说，N(R) 可以告诉远端 LLC，前 N(R)－1 个 I PDU 都已经收到，现在可以发送第 N(R) 个 I PDU 了。

DSAP	PTYPE	SSAP	Sequence(序列)		信息
DDDDDD	1100	SSSSSS	N(S)	N(R)	服务数据单元(SDU)

7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	…	7|6|5|4|3|2|1|0
字节偏移0	字节偏移1	字节偏移2	字节偏移3	…	字节偏移n-1

图 5-16　I PDU 格式

5.2.2.11　RR

RR（Receive Ready）PDU 用于确认一个或多个收到的 I PDU，并指示远端 LLC 可以发送接下来的 I PDU，其消息格式如图 5-17 所示。在 RR PDU 的 Sequence 域中，N(S) 没有用处，设置为 0。N(R) 表示已经收到了编号为 N(R)－1 的 I PDU。RR 的 N(R) 和 I PDU 中的 N(R) 的作用是一样的。RR 专门用于确认收到的 I PDU，而 I PDU 中的 N(R) 是当有 I PDU 要发送时才确认接收到的 I PDU。

DSAP	PTYPE	SSAP	Sequence(序列)	
DDDDDD	1101	SSSSSS	0000	N(R)

7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0
字节偏移0	字节偏移1	字节偏移2

图 5-17　RR PDU 格式

DSAP	PTYPE	SSAP	Sequence(序列)	
DDDDDD	1110	SSSSSS	0000	N(R)

7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0
字节偏移0	字节偏移1	字节偏移2

图 5-18　RNR PDU 格式

5.2.2.12　RNR

RNR（Receive Not Ready）PDU 用于指示 LLC 目前不能继续接收 I PDU，其消息数据格式如图 5-18 所示。

N(R) 表示已经收到了编号为 N(R)－1 的 I PDU，第 N(R) 个 I PDU 将无法接收。

5.2.2.13　SNL

SNL（Service Name Lookup）PDU 用于本地 LLC 确认某个服务在远端 LLC 上是否可用，其消息格式如图 5-19 所示。SNL 的信息域中如果包含 SDREQ 参数，表示服务发现请求消息，SDREQ 参数中包含要请求服务的名称。远端 LLC 收到 SNL PDU 后，通过服务名

称找到对应的 SAP，然后回复 SNL PDU。该回复的 SNL PDU 中包含 SDRES 参数，用于指示服务名称对应的 SAP。SNL PDU 的 SSAP/DSAP 设置为 00001，表示由 SDP 发送和接收。

DSAP	PTYPE	SSAP	信息
000001	1001	000001	参数表

7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	7|6|5|4|3|2|1|0	…	7|6|5|4|3|2|1|0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n-1

图 5-19　SNL PDU 格式

5.2.3　LLC PDU 参数

5.2.3.1　参数描述

PAX、CONNECT、CC 和 SNL 这四个 LLC PDU 中可以包含参数。每个参数是以 TLV（Type/Length/Value，类型/长度/值）的方式封装，其格式如图 5-20 所示。

Type(类型)	Length(长度)	Value(值)		
××h	*n*	第1个字节	…	第*n*个字节
7　　0	7　　0	7　　0	…	7　　0
字节偏移0	字节偏移1	字节偏移2	…	字节偏移n+1

图 5-20　参数 TLV 格式

参数列表如表 5-4 所示。

表 5-4　参数列表

参　　数	包含选项	所包含的 PDU
VERSION，版本号	应当包含	PAX
MIUX（Maximum Information Unit Extension，最大信息单元扩展）	可以包含	PAX，CONNECT，CC
WKS（Well-Known Service List，已知服务列表）	应当包含	PAX
LTO（Link Timeout，链路超时）	可以包含	PAX
RW（Receive Window，接收窗口）	可以包含	CONNECT，CC
SN（Service Name，服务名称）	可以包含	CONNECT
OPT（Option，选项）	可以包含	PAX
SDREQ（Service Discovery Request，服务发现请求）	可以包含	SNL
SDRES（Service Discovery Response，服务发现响应）	可以包含	SNL

5.2.3.2　VERSION

在 LLC 链路激活过程中，需要协商版本号。版本号携带在 PAX PDU 中，版本的编码如图 5-21 所示。基于 LLCP 规范 1.1，主版本号 Major 和次版本号 Minor 均设置为 01h。

5.2.3.3　MIUX

MIUX 称为最大信息单元扩展。MIU 表示 LLC PDU 的最大长度，在 LLCP 规范中默认为 128 字节。LLCP 提供了对于最大信息单元的扩展功能，如果 LLC 支持 MIUX，MIU

的长度为：MIU＝128＋MIUX。MIUX 的格式如图 5-22 所示。MIUX 可以包含在 PAX、CONNECT、CC PDU 中。

Type(类型)	Length(长度)	Value(值)	
01h	01h	Major	Minor
7　0	7　0	7　4	3　0
字节偏移0	字节偏移1	字节偏移2	

图 5-21　版本编码

5.2.3.4　WKS

WKS（Well-Known Service）是一个配置参数，用于表示 LLC 是否支持已知类型的 SAP，其格式如图 5-23 所示。已知类型 SAP 的取值范围为 00h～0Fh，共 16 个已知类型的 SAP。WKS 由 16bit 组成，每一位对应一个已知类型的 SAP。最低位默认设置为 1，表示默认支持链路管理功能的 SAP，即 SAP 等于 0 的情况。WKS 参数包含在 PAX PDU 中。

Type(类型)	Length(长度)	Value(值)	
02h	02h	00000b	MIUX
7　0	7　0	15　11	10　0
字节偏移0	字节偏移1	字节偏移2	字节偏移3

图 5-22　MIUX 格式

Type(类型)	Length(长度)	Value(值)	
03h	03h	XXXXXXXXXXXXXXX1b	
7　0	7　0	F E D C B A 9 8	7 6 5 4 3 2 1 0
字节偏移0	字节偏移1	字节偏移2	字节偏移3

图 5-23　WKS 格式

5.2.3.5　LTO

LTO（Link TimeOut）参数规定了从本地 LLC 发出 LLC PDU 到远端 LLC 收到 LLC PDU 之间最大的时间间隔。LTO 由 8bit 组成，表示的时间为 10ms 的整数倍，如图 5-24 所示。

5.2.3.6　RW

RW（Receive Window）称为接收窗口，包含在 CONNECT PDU 或 CC PDU 中，用于表示本地 LLC 的数据链路连接上所能接收不经过确认的 I PDU 个数，如图 5-25 所示。举例来说，RW 设置为 0，表示本地 LLC 数据链路连接不能接收 I PDU；如果 RW 设置为 1，表示本地 LLC 将确认每一个收到的 I PDU，在确认之前，不会接收额外的 I PDU。

Type(类型)	Length(长度)	Value(值)
04h	01h	LTO
7　0	7　0	7　0
字节偏移0	字节偏移1	字节偏移2

图 5-24　LTO 格式

Type(类型)	Length(长度)	Value(值)	
05h	01h	0000b	RW
7　0	7　0	7 6 5 4	3　0
字节偏移0	字节偏移1	字节偏移2	

图 5-25　RW 格式

5.2.3.7　SN

SN（Service Name）可以包含在 CONNECT PDU 中，发送给 SDP 的 SAP（SAP＝1），用于表示与注册在远端 LLC 上的服务建立连接。该服务的服务名即 SN 中表示的 URI，如图 5-26 所示。

对于已知类型的服务名来说，格式为 urn：nfc：sn：<servicename>；对于非已知类型的服务名来说，格式为 urn：nfc：xsn：<domain>：<servicename>。

5.2.3.8　OPT

OPT（Opiton）参数一般用于表示 LLC 支持的链路服务类型 LSC（Link Service Class），如图 5-27 所示，分别表示链路类型 1、2、3 及未知。

Type(类型)	Length(长度)	Value(值)	
06h	*n*	服务名称URI	
7　0	7　0	7　0	… 7　0
字节偏移0	字节偏移1	字节偏移2	… 字节偏移n+1

图 5-26　SN 格式

Tpye(类型)	Length(长度)	Value(值)	
07h	01h	000000b	LSC b_1b_0
7　0	7　0	7 6 5 4 3 2	1 0
字节偏移0	字节偏移1	字节偏移2	

图 5-27　OPT 格式

5.2.3.9　SDREQ

SDREQ（Service Discovery Request）可以包含在 SNL PDU 中，发送给 SDP 的 SAP。SDREQ 中包含服务名称 URI，用于请求远端 LLC 返回服务名称 URI 所对应的 SAP，如图 5-28 所示。其中，TID（Transaction Identifier）用于关联服务发现请求和服务发现响应，即请求和响应必须使用相同的 TID。服务名称 URI 的格式与 SN 参数中的一致。

Tpye(类型)	Length(长度)	Value(值)		
08h	*n*	TID	服务名称URI	
7　0	7　0	7　0	7　0	… 7　0
字节偏移0	字节偏移1	字节偏移2	字节偏移3	… 字节偏移n+1

图 5-28　SDREQ 格式

5.2.3.10　SDRES

SDRES（Service Discovery Response）可以包含在 SNL PDU 中，用于响应包含 SDREQ 的 SNL PDU，如图 5-29 所示。其中，TID 与 SDREQ 中的 TID 含义一致。SAP 值与 SDREQ 中的服务名称对应。当本地 LLC 收到远端 LLC 发送的包含 SDRES 参数的 SNL PDU 后，就可以根据该 SAP 值建立连接或发送 PDU。

Tpye(类型)	Length(长度)	Value(值)	
09h	02h	TID	0 0 SAP
7　0	7　0	7　0	7 6 5　0
字节偏移0	字节偏移1	字节偏移2	字节偏移3

图 5-29　SDRES 格式

5.3　LLC 工作流程

5.3.1　链路激活

在点对点通信模式下，MAC 层激活后，需要激活 LLC 链路。MAC 层的激活过程在第 4 章已经介绍，即 NFC-DEP 协议中通过 ATR_REQ/ATR_RES 进行参数协商的过程。LLC 的激活是通过在本地 LLC 和远端 LLC 之间交换 PAX PDU 进行参数协商。这些参数包括 LLCP 版本号、MIU、MIUX、WKS、LTO 和 OPT。

需要注意的是，LLC 激活所需的部分参数也可以在 MAC 层激活的过程中传递，如通过 ATR_REQ/ATR_RES 中的通用字节来携带 LLC 参数。如果 ATR_REQ/ATR_RES 支

持携带 LLC 参数，那么在 LLC 激活中，只需要通过 PAX PDU 协商 LLCP 版本号和 MIU。如果不允许 ATR _ REQ/ATR _ RES 携带 LLC 参数，那么在 LLC 的激活中，需要通过 PAX PDU 携带所有相关参数，包括版本号和 MIU。LLC 链路激活流程如图 5-30 所示。

LLCP 版本协商用于判断本地 LLCP 版本和远端 LLCP 版本是否匹配和兼容。如果二者的主版本号和次版本号均一致，那么版本协商成功，可以继续进行 MIU 协商。如果主版本号一致，而次版本号不一致，那么版本协商成功，可以继续进行 MIU 协商，最终结果以版本号低的为准。如果主版号不一致，则由主版本号高的一方决定版本协商是否成功，即由高版本号的 LLCP 来判断是否兼容低版本号的 LLCP。

MIU 协商在版本协商之后进行。MIU 一般默认为 128 字节。如果本地 LLC 能够接收大于 128 字节的信息域，那么在 PAX PDU 中包含 MIUX 参数。如果本地 LLC 只支持 128 字节信息域，将不发送 MIUX 参数。

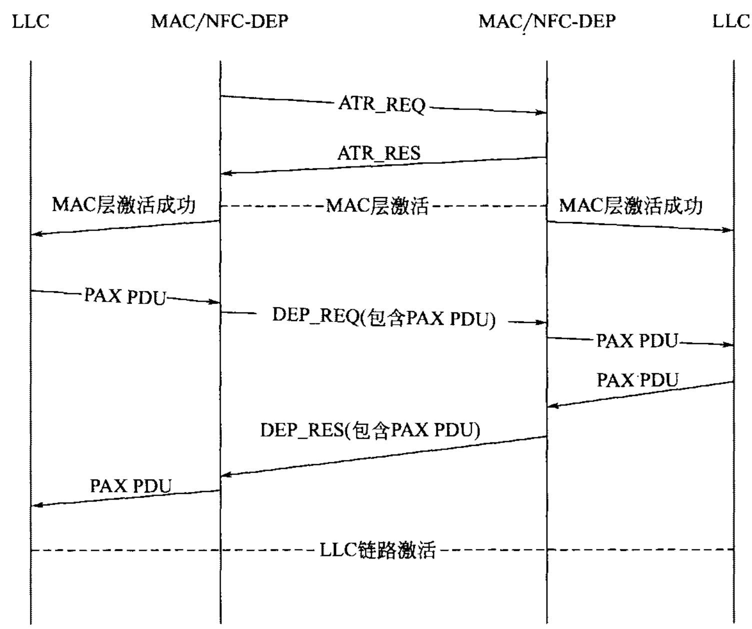


图 5-30 LLC 链路激活流程

在 LLC 链路激活后，就可以使用 LLC 提供的数据传输服务，如发送和接收无连接的 UI PDU；可以通过 CONNECT 建立连接并收发 I PDU，发送、接收 AGF PDU，通过 SYMM PDU 维护 LLC 链路，执行服务发现过程，执行链路的去激活等流程。

5.3.2 链路去激活

LLC 链路的去激活用于结束本地 LLC 和远端 LLC 之间的数据传输服务。除了通过主动发送/接收 DISC PDU 来进行 LLC 链路去激活外，链路超时，即在 LTO 定义时间内没有收到任何 PDU 的情况下，链路也会被去激活。LLC 链路去激活的流程如图 5-31 所示。

5.3.3 面向连接的流程

在面向连接的服务中，首先需要建立连接。LLC 发送 CONNECT PDU 请求建立连接，在 CONNECT PDU 中包含连接相关的参数。在本地 LLC 未收到远端 LLC 发送的 DM PDU 或 CC PDU 之前，本地 LLC 将不会发送相同 SSAP 的 CONNECT PDU。如果远端 LLC 接

受了 CONNECT PDU 中的连接参数，将回复 CC PDU，确认建立连接；否则，发送 DM PDU，表示拒绝连接请求。当收到 CC PDU 后，LLC 把 V(S)、V(R)、V(SA) 和 V(RA) 初始化为 0，再发送和接收 I PDU。面向连接服务的连接建立过程如图 5-32 所示。

在面向连接的服务中，I PDU、RR PDU 和 RNR PDU 中都包含了发送序列号和接收序列号，因此 LLCP 中定义了一些状态变量和参数来描述面向连接服务的工作流程。需要注意的是，这些状态变量只是为了更好地描述面向连接服务的工作流程，并不作为强制的规定使用在硬件或软件实现中。

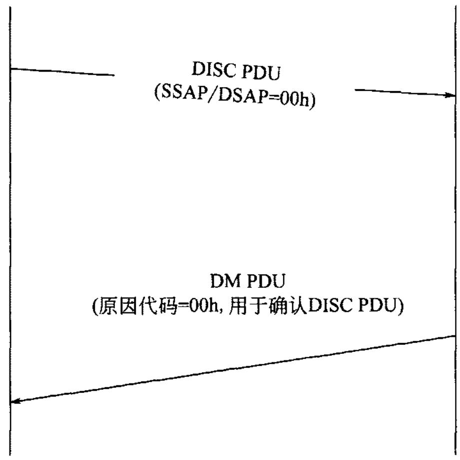


图 5-31 LLC 链路去激活流程

V(S) 称为发送状态变量 (Send State Variable)，等于下一个即将发送的 I PDU 序列号做模 16 运算，即 V(S)＝N(S)/16，取值范围为 0～15。V(SA) 称为发送确认状态变量 (Send Acknowledgement State Variable)，等于最新收到的 I PDU 的 N(R)。V(R) 称为接收状态变量 (Receive State Variable)，等于下一个即将接收的 I PDU 序列号做模 16 运算，即 V(R)＝N(R)/16，取值范围为 0～15。V(RA) 称为接收确认状态变量 (Receive Acknowledgement State Variable)，等于最新发送的 I PDU 的 N(R)。

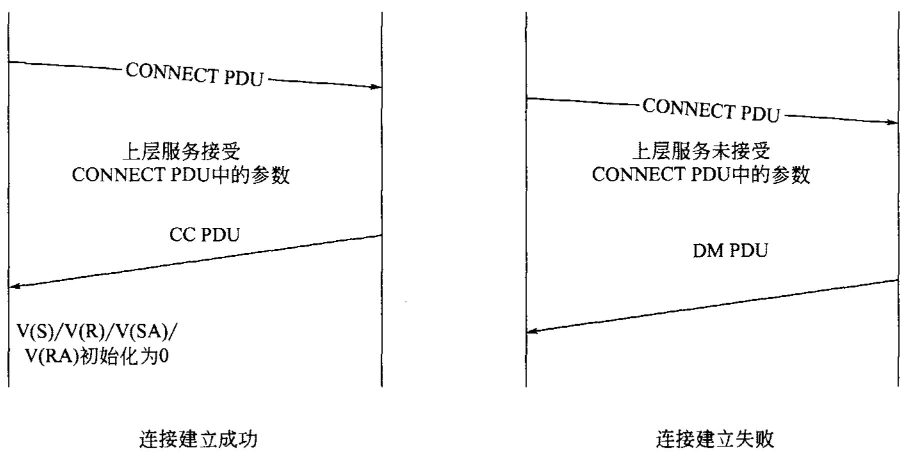


图 5-32 面向连接服务的连接建立过程

另外，面向连接的服务中还定义了两个参数 RW(L) 和 RW(R)。RW(L) 称为本地接收窗口 (Local Receive Window)，表示本地 LLC 能够接收远端 LLC 的 I PDU，且无需确认的 I PDU 的个数，取值范围为 1～15。举例来说，如果 RW(L) 取值为 1，表示本地 LLC 每次只能接收一个 I PDU，因此远端 LLC 一次只能向本地 LLC 发送一个 I PDU，且必须得到本地 LLC 的确认后，才能发送下一个 I PDU。如果 RW(L) 设置为 5，表示远端 LLC 可以连续向本地 LLC 发送 5 个 I PDU，且无需确认。RW(L) 可以包含在 CONNECT 或 CC PDU 中。RW(R) 称为远端接收窗口 (Remote Receive Window)，表示远端 LLC 所能接收无需确认 I PDU 的个数，其具体含义与 RW(L) 一样。RW(R) 可以包含在CONNECT或

CC PDU 中。RW(L) 和 RW(R) 实际上是相对的概念，一个表示本地 LLC 的接收窗口，另一个表示远端 LLC 的接收窗口。

面向连接服务的 I PDU 接收和发送遵循以下规则：

① 当 V(S)＝V(SA)＋RW(R) 时，LLC 将不发送任何 I PDU。当 LLC 发送一个 I PDU时，设置 N(S) 值等于 V(S)，N(R) 等于 V(R)。发送完 I PDU 之后，V(S) 加 1。

② 若 LLC 收到一个 I PDU，并且其 N(S) 等于 V(R)，LLC 将 I PDU 信息域中的内容上报给上层服务，并对 V(R) 加 1。

③ 当至少有一个 I PDU 可以被发送时，即将发送的 I PDU 将携带 N(R)，确认收到的 I PDU。如果没有 I PDU 需要发送，并且可以接收更多的 I PDU，将发送 RR PDU 确认收到的 I PDU，否则发送 RNR PDU。

④ 收到 RNR PDU 后，本地 LLC 将停止发送 I PDU。

5.3.4 无连接的流程

无连接的传输服务是指在传输数据 UI PDU 之前无需建立连接，一旦 SSAP/DSAP 确定，就可以收发 UI PDU。UI PDU 不能超过 MIU 的值，并且 UI PDU 不需要确认。

5.3.5 链路维护流程

LLC 链路的维护是通过 Symmetry 流程完成的。Symmetry 流程采用 ABM 通信模式。链路维护流程如下所述：

① 收到 LLC PDU 后，本地 LLC 将在 LTO 规定的时间内向远端 LLC 发送 LLC PDU。如果在 LTO 规定的时间内没有要发送的 LLC PDU，LLC 将发送 SYMM PDU 给远端 LLC。

② 在发送 LLC PDU 后，如果在 LTO 时间内没有收到任何 LLC PDU，本地 LLC 将执行链路去激活。

5.3.6 服务发现流程

服务发现流程允许本地 LLC 根据服务名称从远端 LLC 获取该服务名称对应的 SAP。服务发现请求 SDREQ 参数携带在 SNL PDU 中，服务发现响应 SDRES 参数同样携带在 SNL PDU 中，用于响应携带 SDREQ 参数的 SNL PDU。

收到服务发现请求后，LLC 将确定服务名称是否绑定了某个 SAP。如果存在相应的 SAP，LLC 将 SAP 存放在 SDRES 参数中，通过 SNL PDU 返回给请求服务发现的 LLC。在获取目标 SAP 后，就可以进行面向连接的建立、数据传输或无连接的数据传输。服务发现流程如图 5-33 所示。

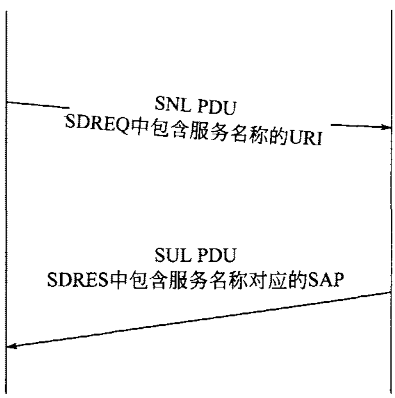


图 5-33 服务发现流程

5.4 MAC 层映射

LLC 层可以被映射到不同的 MAC 层上。在 LLCP 规范中，规定了基于 NFC-DEP 的 MAC 层映射。除了规定将 LLC PDU 封装在 DEP _ REQ/RES 消息中之外，NFC-DEP 还可以为 LLC 提供链路激活参数的交互。LLCP 中定义的部分链路激活的参数可以包含在 ATR _

REQ/RES 消息中，在 MAC 层链路激活的过程中协商 LLC 层链路参数。如果采用这种方式，在 LLC 链路激活过程中只需通过 PAX PDU 交互 VERSION 和 MIU 即可。图 5-34 描述了 ATR _ REQ 消息的通用字节携带的 LLC 参数。其中，“46h 66h 6Dh”称为 LLCP 幻数（Magic Number），表示通用字节中包含 LLCP 的参数 TLV。同样，目标设备收到携带 LLCP 参数的 ATR _ REQ 后，将自己的 LLCP 参数包含在 ATR _ RES 的通用字节中，并以“46h 66h 6Dh”开头。

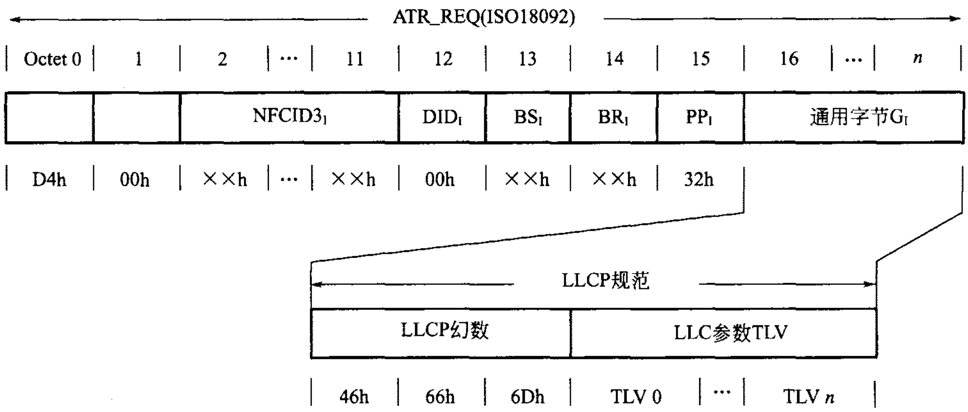


图 5-34　ATR _ REQ 消息携带的 LLC 参数

5.5　小结

本章介绍了 LLCP 协议的相关内容。首先介绍 LLCP 的基本概念，如 LLCP 的协议架构、LLCP 的链路服务类型、面向连接的数据传输服务和无连接的数据服务；然后介绍 LLCP 的 PDU 格式及类型，以及 LLCP 链路激活流程、去激活流程、Symmtery 流程和服务发现流程；最后介绍 MAC 层映射的流程。

第6章 标签操作规范

6.1 标签操作规范概述

NFC Forum 定义了四类标签：类型 1、类型 2、类型 3 和类型 4 标签。类型 1 标签基于 NFC-A 技术，兼容 Innovision Topaz 的非接触卡/标签产品。类型 2 标签基于 NFC-A 技术，兼容恩智浦的 Mifare Ultralight 和 Mifare Ultralight C 等产品。类型 3 标签基于 NFC-F 技术，兼容 Felica 非接触卡/标签产品。类型 4 标签分为 A、B 两种，分别基于 NFC-A 和 NFC-B 技术，兼容 ISO 14443 的非接触卡/标签产品。

为了支持工作在读写模式下的 NFC 设备对四类标签进行读写，NFC Forum 分别定义了四类标签操作规范：类型 1 标签操作规范（Type 1 Tag Operation Specification）[1]、类型 2 标签操作规范（Type 2 Tag Operation Specification）[2]、类型 3 标签操作规范（Type 3 Tag Operation Specification）[3] 和类型 4 标签操作规范（Type 4 Tag Operation Specification）[4]。标签操作规范描述了工作在读写模式下的 NFC 设备如何对四类标签进行操作和管理，同时定义了 NFC 设备如何从四类标签中发现、读取或写入 NDEF 数据。

6.2 类型 1 标签操作规范

6.2.1 存储结构和管理

根据存储容量的大小，类型 1 标签的存储结构分为静态存储结构和动态存储结构两种。静态存储结构应用于物理存储容量为 120 字节的标签，动态存储结构应用于物理存储容量大于 120 字节的标签。

类型 1 标签的存储被分为多个数据块（Block），每一个块由 8 字节组成。在静态存储结构中，包含 15 个块，每一个块都有一个编号，编号为 0～14。在动态存储结构中，包含超过 15 个数据块，编号为 0～k。

---

[1] Type 1 Tag Operation Specification，T1TOP 1.1.

[2] Type 2 Tag Operation Specification，T2TOP 1.1.

[3] Type 3 Tag Operation Specification，T3TOP 1.1.

[4] Type 4 Tag Operation Specification，T4TOP 2.0.

类型 1 标签包含 2 字节长的固定 ROM，称为 HR0 和 HR1，如图 6-1 所示。HR0 的高 4 位取值为 0001，表示该标签为类型 1 的，而且支持存储 NDEF 数据。HR0 的低 4 位取值为 0001，表示该标签为静态存储结构。如果不等于 0001，则为动态存储结构。HR1 暂时未使用，可以取任意值。

HR0	HR1
1×h	××h

图 6-1　HR0 和 HR1

类型 1 标签的静态存储结构如图 6-2 所示。静态存储结构包括 15 个数据块，编号为 0～14(Eh)。

静态存储结构										
类型	块编号	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 (MSB)	是否支持被锁定
UID	0	UID0	UID1	UID2	UID3	UID4	UID5	UID6	预留	锁定
Data	1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	是
Data	2	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	是
Data	3	Data16	Data17	Data18	Data19	Data20	Data21	Data22	Data23	是
Data	4	Data24	Data25	Data26	Data27	Data28	Data29	Data30	Data31	是
Data	5	Data32	Data33	Data34	Data35	Data36	Data37	Data38	Data39	是
Data	6	Data40	Data41	Data42	Data43	Data44	Data45	Data46	Data47	是
Data	7	Data48	Data49	Data50	Data51	Data52	Data53	Data54	Data55	是
Data	8	Data56	Data57	Data58	Data59	Data60	Data61	Data62	Data63	是
Data	9	Data64	Data65	Data66	Data67	Data68	Data69	Data70	Data71	是
Data	A	Data72	Data73	Data74	Data75	Data76	Data77	Data78	Data79	是
Data	B	Data80	Data81	Data82	Data83	Data84	Data85	Data86	Data87	是
Data	C	Data88	Data89	Data90	Data91	Data92	Data93	Data94	Data95	是
预留	D									
锁定/预留	E	LOCK0	LOCK1	OTP0	OTP1	OTP2	OTP3	OTP4	OTP5	

图 6-2　类型 1 标签静态存储结构

数据块 0 存储了 7 字节的 UID（Unique Iddentification)。其中，字节 6 为标签制造商的标识代码，字节 7 为预留字节，字节 0～字节 5 表示唯一的数字。每一个数据块都有是否支持锁定的属性。如果支持锁定属性，该数据块被锁定为只读；如果不支持该属性，表示该数据块不能被锁定，可读也可写。需要注意的是，该属性只表示数据块是否支持被锁定，并不是说该数据块必须被锁定为只读。数据块 0h 默认锁定为可读，因为 UID 唯一地标识一个标签。数据块 1h～Ch 为通用的数据存储块。数据块 Dh 为预留数据块，不存放任何数据，默认锁定为只读。数据块 Eh 也默认锁定为只读，字节 0 和字节 1 用于表示各个数据块是否被锁定，每一个比特位对应一个数据块，如果设置为 1，表示该数据块被锁定为只读，否则表示该数据块可读可写，如图 6-3 所示。字节 2～字节 7 用于表示 OTP（One Time Programmable)，并未被 NFC Forum 使用。

类型 1 标签的动态存储结构如图 6-4 所示。其中，数据块 0h～Eh 与静态存储结构一致。这里需要注意，图 6-4 中的 Fh 和末尾两个块的锁定/预留字节只用于举例。实际上，在动态存储结构中，锁定字节和预留字节可以位于存储结构的 Fh 到结束之间的任意位置，因此需要一些信息来指示锁定字节和预留字节所在的位置。

动态存储结构定义了存储控制 TLV（Memoery Control TLV）来描述预留字节的位置，定义了锁定控制 TLV（Lock Control TLV）来描述锁定字节具体的位置。在静态存储结构

LOCK0 字节0								LOCK1 字节1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	B5	b4	b3	b2	b1	b0
0b=Blcok7 未被锁定	0b=Blcok6 未被锁定	0b=Blcok5 未被锁定	0b=Blcok4 未被锁定	0b=Blcok3 未被锁定	0b=Blcok2 未被锁定	0b=Blcok1 未被锁定	1b=Blcok0 锁定	未被使用	1b=BlcokE 锁定	1b=BlcokD 锁定	0b=BlcokC 未被锁定	0b=BlcokB 未被锁定	0b=BlcokA 未被锁定	0b=Blcok9 未被锁定	0b=Blcok8 未被锁定

图 6-3 数据块 Eh

动态存储结构										
类型	块编号	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 (MSB)	是否支持被锁定
-------数据块0h～Eh-----										
锁定/预留	Fh	LOCK 2	LOCK 3							
Data	10h	Data96	Data97	Data98	Data99	Data100	Data101	Data102	Data103	是
Data	11h	Data104	Data105	Data106	Data107	Data108	Data109	Data110	Data111	是
Data	12h	Data112	Data113	Data114	Data115	Data116	Data117	Data118	Data119	是
Data	13h	Data120	Data121	Data122	Data123	Data124	Data125	Data126	Data127	是
Data	14h	Data128	Data129	Data130	Data131	Data132	Data133	Data134	Data135	是
Data	15h	Data136	Data137	Data138	Data139	Data140	Data141	Data142	Data143	是
Data		Data144	Data145	Data146	Data147	Data148	Data149	Data150	Data151	是
Data		Data152	Data153	Data154	Data155	Data156	Data157	Data158	Data159	是
Data		Data160	Data161	Data162	Data163	Data164	Data165	Data166	Data167	是
Data		Data168	Data169	Data170	Data171	Data172	Data173	Data174	Data175	是
Data		Data176	Data177	Data178	Data179	Data180	Data181	Data182	Data183	是
Data		Data184	Data185	Data186	Data187	Data188	Data189	Data190	Data191	是
锁定/预留		LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	
锁定/预留	k	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	LOCK x	

图 6-4 动态存储结构

中，存储大小固定，Dh的数据块表示预留字节，Eh的数据表示锁定字节，因此无需存储控制TLV和锁定控制TLV。一般情况下，锁定控制TLV和存储控制TLV位于CC（Capability Container）之后，CC用于表示类型1标签中的相关信息，如标签中是否包含NDEF消息，类型1标签的版本信息，类型1标签的物理存储容量等信息。CC位于数据块1的前4个字节。在动态存储结构中，CC之后是5字节的锁定控制TLV和5字节的存储控制TLV，如图6-5所示。锁定控制TLV位于数据块1h的字节5～字节7和数据块2h的字节0，存储控制TLV位于数据块2h的字节1～字节5。

CC的结构如图6-6所示。NMN取值为E1h，表示标签中包含NDEF消息。NDEF消息的内容也是通过TLV的形式存储。VNo为标签支持的版本号，高4位表示主版本号，低4位表示次版本号，10h表示版本1.0。TMS表示标签物理存储容量，静态存储结构为0Eh个8字节。RWA表示读写访问属性，高4位表示读属性，默认为0h，表示有读权限；低4

动态存储结构										
类型	块编号	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 (MSB)	是否支持被锁定
UID	0	UID 0	UID 1	UID 2	UID 3	UID 4	UID 5	UID 6	预留	锁定
Data	1	CC				锁定控制TLV				是
Data	2		存储控制TLV					Data14	Data15	是

图 6-5　动态存储结构中锁定控制 TLV 和存储控制 TLV 的位置

位表示写属性，0h 表示有写权限，Fh 表示无写权限。

TLV 由类型、长度和值组成，其中类型如表 6-1 所示。长度有两种编码格式：1 字节长度和 3 字节长度，如图 6-7 所示。当长度取值为 00h～FEh 时，表示 1 字节长度；长度值超过 FEh 时，为 3 字节长度。其中，第一个字节取值为 FFh，表示后面还有 2 字节表示真正的长度，取值为 00FFh～FFFEh。

Byte 0	Byte 1	Byte 2	Byte 3
NMN, NDEF “幻数” (E1h)	VNo, 版本号 (10h)	TMS, 标签存储容量	RWA, 读写访问

图 6-6　CC 结构

00～FE	1字节长度

FF	00FF～FFFE	3字节长度

图 6-7　字节长度

除了存储控制 TLV、锁定控制 TLV 之外，类型 1 标签还定义了其他 TLV，如表 6-1 所示。

表 6-1　TLV 定义

TLV 名称	类　型	描　述	作　用　域
NULL TLV	00h	用于填充存储域	静态存储结构、动态存储结构
锁定控制 TLV	01h	描述锁定字节信息	动态存储结构
存储控制 TLV	02h	描述预留字节信息	动态存储结构
NDEF 消息 TLV	03h	描述 NDEF 消息	静态存储结构、动态存储结构
私有 TLV	FDh	描述私有信息	静态存储结构、动态存储结构
结束 TLV	FEh	数据结束的 TLV	静态存储结构、动态存储结构

NDEF 消息存放在 NDEF TLV 中。在静态存储区中，如果包含 NDEF 消息，则 NDEF TLV 紧随 CC 之后；在动态存储区中，如果包含 NDEF 消息，则 NDEF TLV 紧随存储控制 TLV 之后。如果有私有 TLV，则私有 TLV 一般紧随 NDEF TLV 之后。数据最后存放结束 TLV，表示数据结束。NULL TLV 和结束 TLV 只有 1 字节，表示类型，没有长度和值。锁定控制 TLV 和存储控制 TLV 一共 5 字节，包括 1 字节的类型、1 字节的长度和 3 字节的值。

锁定控制 TLV 中的值由 3 个字节表示。其中，最高字节表示锁定字节所在位置，高 4 位表示页地址 PagesAddr（0h＝0，…，Fh＝15），低 4 位表示字节偏移量 ByteOffset（0h＝0，…，Fh＝15）。中间字节表示锁定字节包含的比特数（01h＝1，…，FFh＝255，00h＝256），可以不是 8bit 的倍数。最低字节表示页控制信息 Page Control，包括一页包含多少字节和一个锁定位能够锁定多少字节。页控制信息的低 4 位表示一页有多少字节 BytesPerPage，用 2^n 表示（0h＝预

留，1h=1，…，Fh=15)；高4位表示每个比特能锁定的字节数BytesLockedPerLockBit，用2^n表示（0h=预留，1h=1，…，Fh=15)。因此根据图6-8所示的公式，可以计算出锁定字节的开始位置。

$$\text{ByteAddr}=\text{PageAddr}\cdot 2^{\text{BytesPerpage}}+\text{ByteOffset}$$

图6-8 锁定字节/预留字节的开始位置

存储控制TLV的值与锁定控制TLV中的值类似，也由3个字节表示。其中，最高字节表示预留字节所在位置，高4位表示页地址PagesAddr（0h=0，…，Fh=15)，低4位表示字节偏移量ByteOffset（0h=0，…，Fh=15)。中间字节表示预留字节的个数（01h=1，…，FFh=255，00h=256)，可以不是8bit的倍数。最低字节表示部分页控制信息Partial Page Control，低4位表示一页有多少字节BytesPerPage，用2^n表示（0h=RFU，1h=1，…，Fh=15)，高4位预留。同样，根据图6-8所示公式可以计算出预留字节的开始位置。

6.2.2 命令集

第4章介绍过，运行类型1标签平台的NFC读写器可以通过读写命令来读写类型1标签中的内容。这些命令封装在类型1标签平台定义的帧格式中（与NFC-A标准帧略有不同）发送给类型1标签，类型1标签的响应消息封装在NFC-A标准帧中返回给读写器。类型1标签平台的冲突检测过程与NFC-A不同。类型1标签平台通过发送RID命令进行冲突检测和设备激活。类型1标签响应RID命令，向读写器返回HR0/HR1及UID的内容，让运行类型1标签平台的读写器了解类型1标签的HR0/1信息和UID内容，这些信息将应用在后面介绍的命令中。

类型1标签的静态存储结构和动态存储结构定义了不同的命令集，用于类型1标签的读写。图6-9描述了静态存储结构的命令集。静态存储结构的命令集包括RALL、READ、WRITE-E和WRITE-NE命令。

命令	命令长度(字节)	响应长度(字节)
RALL	9	124
READ	9	4
WRITE-E	9	4
WRITE-NE	9	4

图6-9 静态存储结构命令集

这4个命令的长度均为9字节。RALL用于读取类型1标签中的所有内容。READ表示读取类型1标签中某个地址的1字节内容。WRITE-E（Write with Erase）对标签中的某个字节进行写操作，在操作之前，首先擦除该字节。WRITE-NE（Write No Erase）对标签中的某个字节进行写操作，但是不会先擦除，该命令可以对字节中的某几位置位。这4个命令的格式及响应如图6-10所示，类型编码如图6-11所示。

ADD表示地址操作数，即读写操作的数据位置，如图6-12所示。其中，b7～b4表示数据块的编号，b3～b1表示该数据块中的某一个字节。以READ命令为例，如果ADD中的b7～b4为0010，b3～b1为011，表示读写器要读取类型1标签中第2个数据块的第3个字节。

由于动态存储结构的容量大于120字节，因此动态存储结构定义了能够读/写更多内容的命令，如图6-13所示。其中，RSEG（Read Segment）用于读取一个段的内容，段包括

静态存储结构命令和响应																			
(类型)	命令								响应										
RALL	00h	00h	UID0	UID1	UID2	UID3	CRC1	CRC2	HR0	HR1	UID0	…	…	…	…	…	OTP5	CRC1	CRC2
READ	ADD	00h	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2							
WRITE-E	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2							
WRITE-NE	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2							

图 6-10　静态存储结构命令和响应

命令		命令编码(7bit)						
		msb						lsb
		b7	b6	b5	b4	b3	b2	b1
RALL	00h	0	0	0	0	0	0	0
READ	01h	0	0	0	0	0	0	1
WRITE-E	53h	1	0	1	0	0	1	1
WRITE-NE	1Ah	0	0	1	1	0	1	0

图 6-11　静态存储结构命令类型编码

16 个数据块，共 128 字节。READ8 表示一次读取一个数据块的内容，共 8 字节。WRITE-E8 与静态存储结构中的 WRITE-E 命令类似，但是 WRITE-E8 一次写一个数据块，共 8 字节。WRITE-NE8 与 WRITE-NE 类似，但是 WRITE-NE8 一次写一个数据块，共 8 字节。

地址操作数：ADD 数据块：0h～Eh 字节：0～7							
msb							lsb
b8	b7	b6	b5	b4	b3	b2	b1
0b	静态数据块				字节		

图 6-12　静态存储结构命令的地址编码格式

命令	命令长度(字节)	响应长度(字节)
RSEG	16	131
READ8	16	11
WRITE-E8	16	11
WRITE-NE8	16	11

图 6-13　动态存储结构命令集

图 6-14 描述了动态存储结构的命令和响应结构。其中，ADDS 表示段操作的地址操作数，用于表示读写标签中某个段的数据内容。ADD8 表示数据块的地址操作数，用于表示读写标签中某个块的数据内容。

图 6-15 列出了命令类型的编码格式。

ADDS 表示段地址操作数，即读写操作的段数据位置，如图 6-16 所示。其中，b8～b5 表示段数据的编号，b4～b1 设置为 0。

ADD8 表示数据块的地址操作数，即读写操作的数据块位置，如图 6-17 所示。

6.2.3 NDEF 的发现及访问

类型 1 标签中包含 NDEF 消息，以 NDEF TLV 的格式存储。在静态存储结构中，如果包含 NDEF 消息，则 NDEF TLV 紧随 CC 之后；在动态存储区中，如果包含 NDEF 消息，则 NDEF TLV 紧随存储控制 TLV 之后。

命令															
RSEG	ADDS	00h	00h	00h	00h	00h	00h	00h	00h	UID 0	UID 1	UID 2	UID 3	CRC1	CRC2
READ8	ADD8	00h	00h	00h	00h	00h	00h	00h	00h	UID 0	UID 1	UID 2	UID 3	CRC1	CRC2
WRITE-E8	ADD8	DAT 0	DAT 1	DAT 2	DAT 3	DAT 4	DAT 5	DAT 6	DAT 7	UID 0	UID 1	UID 2	UID 3	CRC1	CRC2
WRITE-NE8	ADD8	DAT 0	DAT 1	DAT 2	DAT 3	DAT 4	DAT 5	DAT 6	DAT 7	UID 0	UID 1	UID 2	UID 3	CRC1	CRC2

响应												
ADDS	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	…	DAT 127	CRC1	CRC2
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		

图 6-14 动态存储结构命令和响应

命令		命令类型编码(7bit)						
		msb						lsb
		b7	b6	b5	b4	b3	b2	b1
RSEG	10h	0	0	0	0	0	0	0
READ8	02h	0	0	0	0	0	1	0
WRITE-E8	54h	1	0	1	0	1	0	0
WRITE-NE8	1Bh	0	0	1	1	0	1	1

图 6-15 动态存储结构命令类型编码

段地址操作数：ADDS 块：0h～Fh							
msb						lsb	
b8	b7	b6	b5	b4	b3	b2	b1
段				0b	0b	0b	0b

图 6-16 动态存储结构命令 ADDS 格式

块地址操作数：ADD8 块：00h～FFh							
msb						lsb	
b8	b7	b6	b5	b4	b3	b2	b1
数据块							

图 6-17 动态存储结构命令 ADD8 格式

HR0＝1xh 表示该类型 1 标签支持 NDEF 数据存储，但是不表示该标签中一定包含 NDEF 数据。如果要确定是否包含 NDEF 消息，需要读取 CC 中的 NMN 字节来判断。运行类型 1 标签平台的读写器通过 RID 命令获取该类型 1 标签的 HR0 字段，从而知道该标签是静态存储结构还是动态存储结构。如果是静态存储结构，通过 READ 命令读取数据块 1 的第一个字节来判断该标签是否包含 NDEF 消息；对于动态存储结构，发送 READ8 命令获取数据块 1 的所有内容，然后读取 NMN 字段来判断是否包含 NDEF 消息。

图 6-18 以存放在静态存储结构中的 URI 类型的 NDEF 消息“www. nfc-forum. org”为例，描述了 NDEF 是如何存储的。数据块 1 中的前 4 个字节存放了 CC，CC 的 NMN 设置为 E1h，表示该标签中包含 NDEF 消息。由于该标签为静态存储结构，因此 NDEF TLV 紧随 CC 之后。字节 4 表示 NDEF TLV 类型，字节 5 表示 NDEF 消息的长度为 17h，之后包含

NDEF 消息。关于 NDEF 消息的格式，将在 7.2 节介绍。NDEF 消息结束后，以结束 TLV（Terminitator TLV）表示。

Block No.	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7 (MSB)	是否支持锁定
0	UID 0	UID 1	UID 2	UID 3	UID 4	UID 5	UID 6		锁定
1	CC0 (NMN) =E1h	CC1 (Vno) =10h	CC2 (TMS) =0Eh	CC3 (RWA) =00h	NDEF TLV 类型：T=03h	NDEF TLV 类型：L=17h	D1	02	是
2	12	'S'	'p'	D1	01	0E	'U'	01	是
3	'n'	'f'	'c'	'-'	'f'	'o'	'r'	'u'	是
4	'm'	'.'	'o'	'r'	'g'	Terminator TLV T=FEh			是
5									是
6									是
7									是
8									是
9									是
A									是
B									是
C									是
D									锁定
E	LOCK 0	LOCK 1	OTP 0	OTP 1	OTP 2	OTP 3	OTP 4	OTP 5	

图 6-18 静态存储结构中的 NDEF 消息

6.3 类型 2 标签操作规范

6.3.1 存储结构和管理

与类型 1 标签类似，类型 2 标签也根据存储容量的大小分为静态存储结构和动态存储结构。静态存储结构应用于物理存储容量为 64 字节的标签，动态存储结构应用于物理存储容量大于 64 字节的标签。类型 2 标签的存储被分为多个数据块（Block），每一个块由 4 字节组成。在静态存储结构中，包含 16 个块，每一个块都有一个编号，为 0～15。在动态存储结构中，超过 15 个数据块，编号为 0～k。256 个连续的数据块（1024 字节或 1KB）可以组成为一个扇区（Sector）。

类型 2 标签静态存储结构如图 6-19 所示。数据块 0、数据块 1 和数据块 2 的字节 0、字节 1 为 Internal 字节，预留给标签厂商。数据块 2 的字节 2 和字节 3 为锁定字节，其中的每一位对应一个数据块，表示该数据块是否被锁定。类型 2 标签的 CC 与类型 1 标签的 CC 一样。

类型 2 标签动态存储结构如图 6-20 所示，前 3 个数据块与静态存储结构的前 3 个数据块一致。注意，图 6-20 中末尾的锁定/预留字节只用于举例。实际上，与类型 1 标签类似，在动态存储结构中，锁定字节和预留字节位于数据块 15 之后到结束之间的任意位置，因此需要一些信息来指示锁定字节和预留字节所在的位置。类型 2 标签也定义了存储控制 TLV

字节号	0	1	2	3	块
UID/Internal	Internal0	Internal1	Internal2	Internal3	0
系列号	Internal4	Internal5	Internal6	Internal7	1
Internal/Lock	Internal8	Internal9	Lock0	Lock1	2
CC	CC0	CC1	CC2	CC3	3
Data	Data0	Data1	Data2	Data3	4
Data	Data4	Data5	Data6	Data7	5
Data	Data8	Data9	Data10	Data11	6
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	15

图 6-19 类型 2 标签静态存储结构

（Memoery Control TLV）来描述预留字节的位置，定义了锁定控制 TLV（Lock Control TLV）来描述锁定字节具体的位置。类型 2 标签的 TLV 与类型 1 标签定义的一致，也包括 NULL TLV、锁定控制 TLV、存储控制 TLV、NDEF 消息 TLV、私有 TLV 和结束 TLV。

字节号	0	1	2	3	块
UID/Internal	Internal0	Internal1	Internal2	Internal3	0
系列号	Internal4	Internal5	Internal6	Internal7	1
Internal/Lock	Internal8	Internal9	Lock0	Lock1	2
CC	CC0	CC1	CC2	CC3	3
Data	Data0	Data1	Data2	Data3	4
Data	Data4	Data5	Data6	Data7	5
Data	Data8	Data9	Data10	Data11	6
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	*n*
Lock/Reserved	..	..	..	..	.
Lock/Reserved	..	..	..	..	.
Lock/Reserved	..	..	..	..	*k*

图 6-20 类型 2 标签动态存储结构

6.3.2 命令集

运行类型 2 标签平台的 NFC 读写器，可以通过读写命令来读写类型 2 标签中的内容。类型 2 标签平台是 NFC-A 技术的一个子集，除了 ACK 和 NACK 响应外，其他命令和响应消息都采用 NFC-A 的标准帧，ACK 和 NACK 是一个包含 4 个数据比特的短帧。类型 2 标签平台采用 NFC-A 的冲突检测流程。冲突检测完成后，读写器发送 READ 或 WRITE 命令，将侦听设备激活。类型 2 标签支持的命令有 READ、WRITE 和 SECTOR SELECT。当读写器向类型 2 标签发送这些命令后，类型 2 标签响应 ACK 或 NACK 消息。

READ命令用于读取类型2标签中的内容。其中，类型代码为30h，表示是READ命令；字节2表示要读取的数据块编号，如图6-21所示。

	Byte 1	Byte 2
READ命令	类型 30h	数据块编号 00h～FFh

图6-21　类型2标签READ命令

类型2标签收到READ命令后，将发送响应消息。如果读取成功，返回16字节数据，即READ命令指定的数据块编号在内的后续3个数据块的内容。举例来说，如果READ命令中的数据块编号为4，标签将返回数据块4、5、6、7这四个数据块共16字节内容。如果发生错误，类型2标签返回NACK响应。

类型2标签使用WRITE命令进行写操作。WRITE命令如图6-22所示。如果写成功，标签返回ACK消息；如果失败，返回NACK消息。

	Byte 1	Byte 2	Byte 3～6
WRITE命令	类型 A2h	数据块编号 00h～FFh	数据

图6-22　类型2标签WRITE命令

可以看出，READ命令和WRITE命令的寻址范围只能在00h～FFh之间。当标签的存储容量大于1024KB时，READ/WRITE命令无法寻址，因此类型2标签平台定义了SECTOR SELECT命令来寻址大于1024KB的存储空间。SECTOR SELECT命令分为两个命令包：命令包1和命令包2。命令包1的格式如图6-23所示，包含2个字节。字节1表示类型，即该命令为SECTOR SELECT；字节2设置为FFh。

	Byte 1	Byte 2
SECTOR SELECT 命令包1	类型 C2h	FFh

图6-23　类型2标签SECTOR SELECT命令包1

读写器首先发送命令包1。当类型2标签的存储小于1024KB时，将回复NACK消息，读写器就不再发送命令包2了。如果类型2标签回复ACK命令，表明类型2标签存储大于1024KB，READ/WRITE将无法寻址到大于1024KB的存储空间，因此读写器发送命令包2进行扇区选择。命令包2的格式如图6-24所示。其中，字节1表示所选的扇区号，字节2～

	Byte 1	Byte 2～4
SECTOR SELECT 命令包2	扇区号 00h～FEh, FFh为预留	预留

图6-24　类型2标签SECTOR SELECT命令包2

字节 4 为预留字节。当类型 2 标签收到命令包 2 后，将自己的寻址空间定位至该扇区；读写器发送 READ/WRITE 命令，在该扇区中寻址。

6.3.3 NDEF 的发现及访问

与类型 1 标签类似，类型 2 标签也通过 CC 确认标签内是否存放有 NDEF 消息。图 6-25 举例描述了存放有 NDEF 消息的动态存储结构。从图 6-25 可以看出，CC 之后跟随 5 字节存储控制 TLV，用于表示哪些数据块为预留；存储控制 TLV 之后跟随 5 字节锁定控制 TLV，用于表示哪些数据处于锁定状态，即只能读不能写。锁定控制 TLV 后跟随 NDEF TLV，内容为空。在最后数据结束的位置为结束 TLV。

字节号	0	1	2	3	块
UID/Internal	Internal0	Internal1	Internal2	Internal3	0
系列号	Internal4	Internal5	Internal6	Internal7	1
Internal/Lock	Internal8	Internal9	Lock0=00h	Lock1=00h	2
CC	CC0=E1h	CC1=10h	CC2=0Ch	CC3=00h	3
Lock Control TLV	LockControlTLV0 =01h	LockControlTLV1 =03h	LockControlTLV2 =E0h	LockControlTLV3 =06h	4
Lock Control TLV/ Memory Control TLV	LockControlTLV4 =33h	MemoryControl TLV0=02h	MemoryControl TLV1=03h	MemoryControl TLV2=E1h	5
Memory Control TLV/ NDEF Message TLV	MemoryControl TLV3=0Fh	MemoryControl TLV4=03h	NDEFMessage TLV0=03h	NDEFMessage TLV1=00h	6
Terminator TLV/Data	TerminatorTLV0 =FEh	Data13	Data14	..	7
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	Data93	Data94	Data95	27
Lock/Reserved	Lock2=00h	Reserved0	Reserved1	Reserved2	28
Reserved	Reserved3	Reserved4	Reserved5	Reserved6	29
Reserved	Reserved7	Reserved8	Reserved9	Reserved10	30
Reserved	Reserved11	Reserved12	Reserved13	Reserved14	31

图 6-25 类型 2 标签动态存储结构举例

6.4 类型 3 标签操作规范

6.4.1 存储结构和管理

类型 3 标签与 JIS X6319 规范定义的 Felcia 卡兼容，因此在介绍类型 3 标签之前，先简单介绍 Felica 卡的存储结构和管理。

Felica 卡包含一个或多个系统（System），每一个系统可以看成是一个虚拟的卡，这些虚拟的卡共同存在于物理卡中，如图 6-26 所示。每一个系统有自己的独立存储空间、文件系统、NFCID2 和系统代码（System Code），系统之间相互独立。

一个Felica卡内包含多个虚拟卡，这些虚拟卡的功能不尽相同。比如，一个物理的Felica卡可以用于公交刷卡、地铁刷卡、购买商品、门禁等。系统代码用来标识负责管理系统数据的服务提供商，比如公交、地铁、银行等。在NFC-F的设备发现过程中，尽管所有的目标设备，包括卡/标签，都会响应SENS _ REQ命令，但是如果一个Felcia卡中包含多个系统，只有第一个系统返回SENS _ RES消息，其他系统不会回复。所以，将系统代码设置为FFFFh。因为FFFFh类似于IP网络的广播地址，可以让所有目标设备，包括一个卡中的所有系统都回应。系统代码具有过滤功能，若将系统代码设置为12FCh，那么只有包含NDEF消息的系统才回复。系统代码12FCh分配给NFC Forum用于存放NDEF消息，AA00h～AAFEh分配给ISX6319-4标准，FE00h一般用于通用领域，其他值均由Sony公司管理。

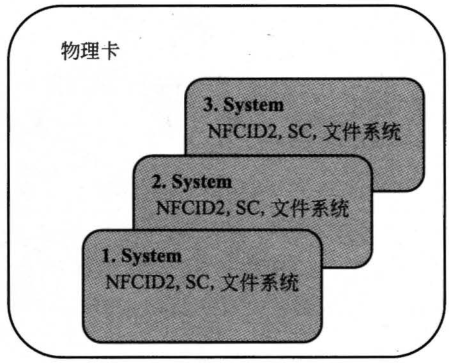


图6-26 物理卡与虚拟卡

Felica文件系统包括四个部分：系统（System）、区域（Area）、服务（Service）和块数据（Block Data）。这四个部分组合在一起管理Felica文件系统。其关系如图6-27所示。

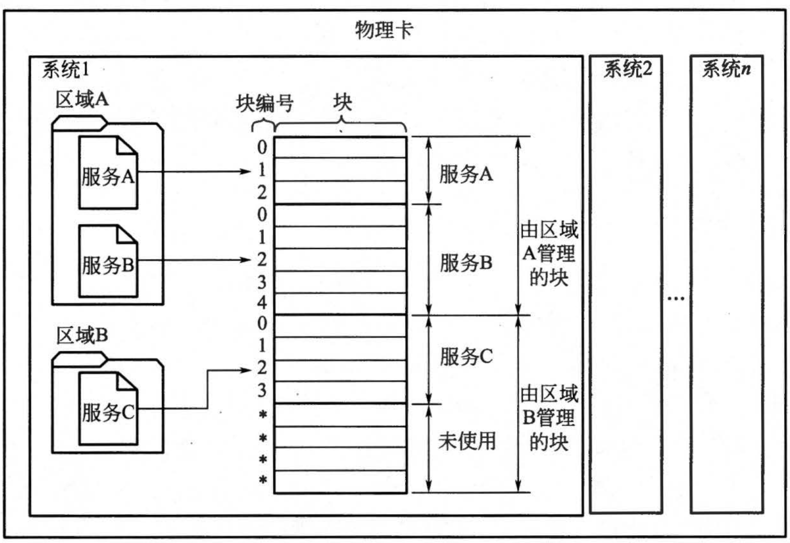


图6-27 Felica卡存储结构与管理

类型3标签与Felica兼容，采用Felcia的存储结构及管理方式。类型3标签以块的形式对存储进行管理。每一个块的大小为16字节。与类型1标签和类型2标签不同，这些数据块不能被直接寻址，也就是说，不能像类型1标签和类型2标签操作规范中定义的那样通过地址直接对数据进行读写操作。类型3标签的数据读写要通过这些数据所属的服务(Service）来操作。

服务类似于文件系统中的文件，每一个服务都关联一组数据块。服务可以通过服务代码(Service Code）寻址，服务代码在类型3标签中是唯一的。类型3标签内包含一些系统信息，由厂家信息、系统定义信息和服务定义信息三部分组成。其中，厂家信息和系统定义信息由类型3标签的生产厂家预先分配。

① 厂家信息如图 6-28 所示，一共 16 字节，包括 IDm 和 PMm 两部分。其中，IDm（Manufacture ID）为 8 字节，用于对类型 3 标签的寻址。实际上 IDm 就是第 4 章介绍的 NFCID2。PMm（Manufacture Parameter）包含最大响应时间参数，PMm 就是第 4 章介绍的 SENSF _ RES 消息中的 PAD0、PAD1、$MRTI_{CHECK}$、$MRTI_{UPDATE}$和 PAD2。

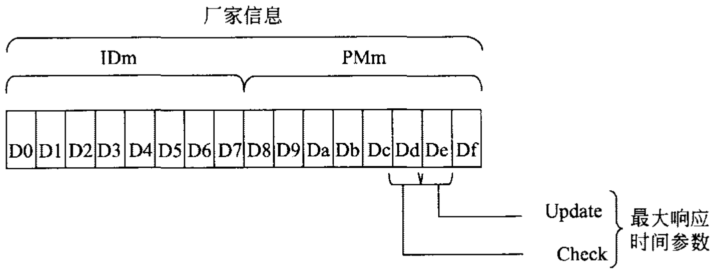


图 6-28 厂家信息

② 系统定义消息就是前面提到的系统代码，由 2 字节组成。

③ 服务定义信息代表类型 3 标签中的每一个服务，包括服务代码（Service Code）和与该服务相关的数据块。服务代码唯一地标识类型 3 标签中的服务，长度为 2 字节，包括服务编号（Service Number）和访问属性（Access Attribute）。服务编号长度为 10bit，唯一地标识一个服务；访问属性长度为 6bit，表示所关联数据块的读写权限。如果访问属性设置为 001001b，表示该服务关联的数据可读可写；如果设置为 001011b，表示该服务关联的数据块只读，不能被写。数据块的编号为 2 字节，用于标识与该服务项关联的数据块。每一个服务定义信息会关联一组数据块。通常情况下，这些数据块完全由所关联的服务使用，但是也有例外，那就是交叠的服务，它们会共享相同的数据块，但是有不同的访问属性。

6.4.2 命令集

类型 3 标签操作规范中定义了三个命令，Polling、Check 和 Upadate。其中，Polling 命令就是第 4 章提到的 SENSF _ REQ 命令，只不过在 Felica 中称为 Polling，二者是一样的，用于设备发现过程。Check 为读取类型 3 标签数据的命令，Update 用于类型 3 标签数据的写操作。本节主要介绍 Check 命令和 Update 命令。

类型 3 标签操作规范的命令编码格式如图 6-29 所示。命令代码标识一个命令。Check 命令代码为 06h，Update 的命令代码为 08h。命令代码之后为该命令中携带的参数。

类型 3 标签操作规范的命令响应消息编码格式如图 6-30 所示。响应代码用于表示响应消息的类型。Check 命令的响应消息代码为 07h，Update 命令的响应消息代码为 09h。

Byte 1	**Byte 2..×**
命令代码	命令参数

图 6-29 类型 3 标签命令编码格式

Byte 1	**Byte 2..×**
响应代码	响应参数

图 6-30 类型 3 标签响应消息编码格式

Check 命令用于从类型 3 标签中读取数据，命令参数如图 6-31 所示。

Check 命令参数中的 IDm 用于表示对哪个标签进行读取操作。服务的个数用于表示读写器在一个命令中准备读取多少个服务的数据内容，一般该值的最大范围由类型 3 标签的实现来决定。服务代码列表中包含一个或多个服务代码，每个服务代码的长度为 2 字节。如果服务的个数为 m，服务代码列表的长度为 $2m$ 个字节。数据块的个数表示读写器在一个命令

参数	长度(字节)	值	说明
IDm	8		厂商相关信息
服务的个数	1	n	需要读取的服务的个数
服务代码列表	$2n$		服务代码的列表，由于每一个服务为2字节，因此长度为$2n$
数据块的个数	1	m	需要读取的数据块的个数
数据块列表	$2m$～$3m$		数据块列表由数据块列表元素组成，每个元素2字节或3字节

图 6-31　Check 命令参数

中准备读取多少个数据块，一般该值的最大范围由类型 3 标签的实现来决定。数据块列表由数据块列表元素组成，数据块列表元素的长度为 2 字节或 3 字节，如图 6-32 所示。其中，字节 0 的 b7 表示长度。如果设置为 0b，表示该数据块列表元素的长度为 2 字节；如果设置为 1b，表示该数据块列表元素的长度为 3 字节。

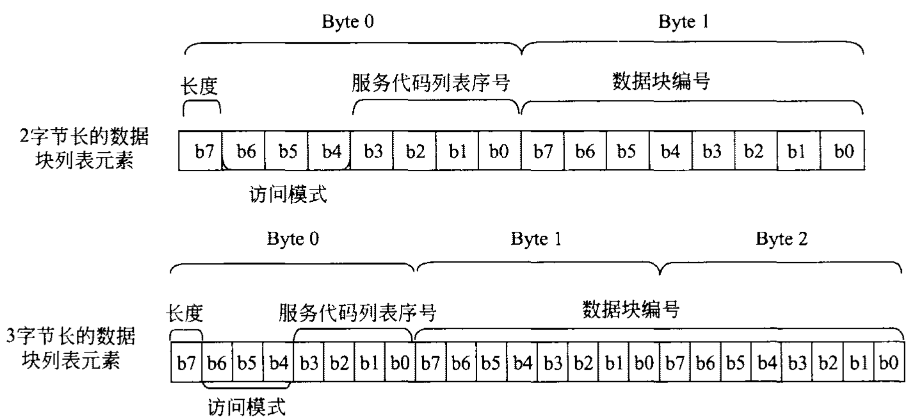


图 6-32　数据块列表元素

字节 0 的 b6～b4 表示对该数据块的访问模式，默认设置为 000b。字节 0 的 b3～b0 表示服务代码列表序号，即服务代码在服务代码列表中的索引。索引从 0 开始计数。举例来说，服务代码列表中包含 3 个服务代码，即 S1、S2 和 S3。如果需要读取服务代码 S3 的某个数据块内容，将数据块列表元素中的服务代码列表序号设置为 0010b，表示要访问服务代码列表中的第三个服务的数据块内容；如果要访问 S1 的内容，需要设置为 0000b。数据块编号可以是 1 字节或 2 字节，表示所对应的服务的数据块编号。类型 3 标签的存储管理结构通过服务关联数据块，因此需要先指定服务，然后才能寻址数据块。

当类型 3 标签收到 Check 命令后，将回复响应消息。Check 命令的响应消息参数如图 6-33所示。

其中，IDm 要与 Check 命令中的 IDm 一致。状态标识 1 和状态标识 2 表示命令的执行情况。状态标识 1 指示错误状态，状态标识 2 用于指示更加详细的错误信息。数据块的个数要与 Check 命令中的一致，数据块类表为返回的数据，长度为 16m 字节。需要注意的是，数据块列表中数据的顺序与数据库列表中的数据块列表元素的顺序是一致的，保证了读写器能够确定这些数据块属于哪一个服务，以及数据块的编号。

图 6-34 举例说明了 Check 命令和响应的用法。Check 命令中包含 1 个服务，服务代码为 000Bh。Check 命令需要读取服务代码为 000Bh 的服务的 3 个数据块，这 3 个数据块的编

参数	长度(字节)	值	说明
IDm	8		厂商相关信息
状态标识1	1		0：命令执行成功 其他值：命令执行失败
状态标识2	1		0：命令执行成功 70h：存储错误，无法对该存储区域进行写操作 71h：该存储区域超过最大的可写次数
数据块的个数	1	*m*	该值与Check命令中的数据块个数一致
数据块列表	16*m*		返回的数据

图 6-33 Check 命令的响应消息参数

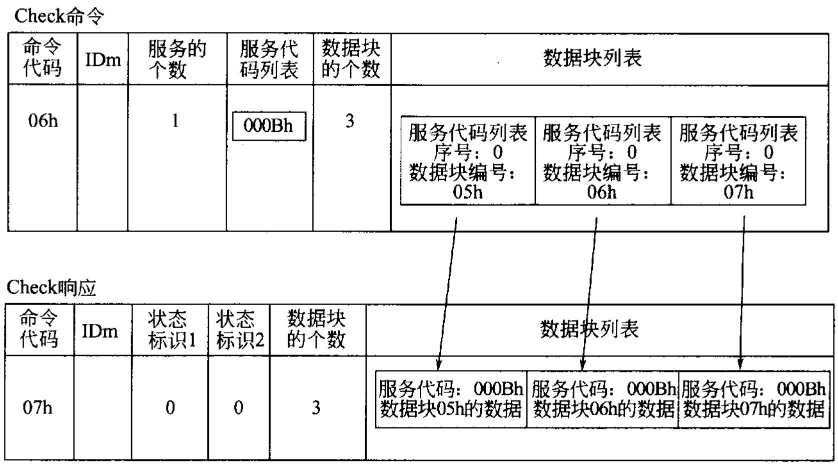


图 6-34 Check 命令和响应的举例

号分别为 05h、06h 和 07h。由于服务代码列表中只有 1 个服务，因此数据块列表中的服务代码列表序号为 0。在 Check 命令的响应中，状态标识 1 和 2 设置为 0，表示 Check 命令执行成功，并返回 3 个数据块的内容，数据块数据的内容按照 05h、06h 和 07h 排列。

Update 命令用于读写器向类型 3 标签写数据。Update 的命令参数如图 6-35 所示。注意，需要写入的数据块数据按照数据块列表中元素的顺序写入。

参数	长度(字节)	值	说明
IDm	8		厂商相关信息
服务的个数	1	*n*	需要读取的服务个数
服务代码列表	2*n*		服务代码的列表，由于每一个服务为2字节，因此长度为2*n*
数据块的个数	1	*m*	需要读取的数据块个数
数据块列表	2*m*～3*m*		数据块列表由数据块列表元素组成，每个元素2字节或3字节
数据块数据	16*m*		需要写入的数据

图 6-35 Update 命令参数

Update 命令的响应消息参数如图 6-36 所示。

6.4.3 NDEF 的发现及访问

前面提到，通过系统代码 12FCh 可以访问到存储在类型 3 标签中的 NDEF 数据。在一

参数	长度(字节)	值	说明
IDm	8		厂商相关信息
状态标识1	1		0：命令执行成功 其他值：命令执行失败
状态标识2	1		0：命令执行成功 70h：存储错误，无法对该存储区域进行写操作 71h：该存储区域超过最大的可写次数

图 6-36　Update 命令的响应消息参数

个系统中，NDEF 数据是由服务代码为 000Bh 的服务来管理，服务 000Bh 的第一个数据块中存放了 NDEF 消息的相关信息，如图 6-37 所示。

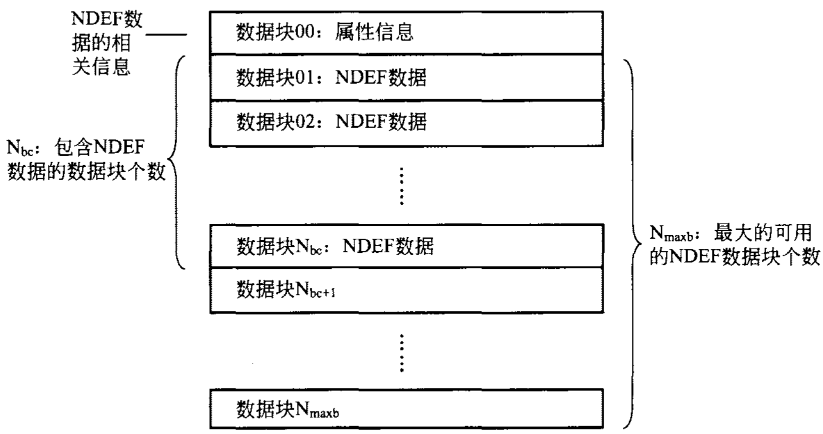


图 6-37　NDEF 数据存储

NDEF 数据的属性信息如图 6-38 所示。

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
V_{er}	N_{br}	N_{bw}	N_{maxb}		unused	unused	unused	unused	WriteF	RW Flag	Ln			Checksum	

图 6-38　NDEF 属性信息

其中，字节 0 表示版本号。字节 1 为 N_{br}，表示 Check 命令所能读取的数据块个数。字节 2 为 N_{bw}，表示 Update 命令所能写的数据块个数。字节 3 和字节 4 为 N_{maxb}，表示可供存放 NDEF 数据的最大数据块个数。字节 5～字节 8 未使用。字节 9 为 WriteF，表示写操作的标志位。设置为 00h，表示写操作已经完成；设置为 0Fh，表示正在进行写操作。NFC 读写器应当在写 NDEF 数据之前，将该标志位设置为 0Fh；写操作完成后，将该值设置为 00h。字节 10 为 RWFlag，表示对于 NDEF 数据的访问属性。00h 表示只读，01h 表示可读可写。字节 11～字节 13 为 Ln，表示存储 NDEF 数据的实际长度，用字节表示，$N_{bc}=Ln/16$。其中，N_{bc}表示包含 NDEF 数据块的个数。

类型 3 标签 NDEF 数据的发现流程如图 6-39 所示。

NDEF 数据的读操作流程如图 6-40 所示。

NDEF 数据的写操作流程如图 6-41 所示。

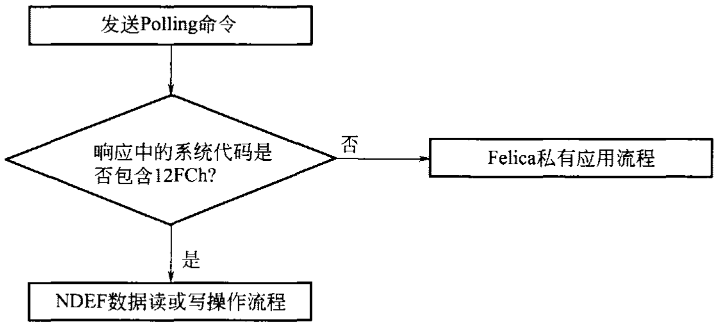


图 6-39 NDEF 数据发现流程

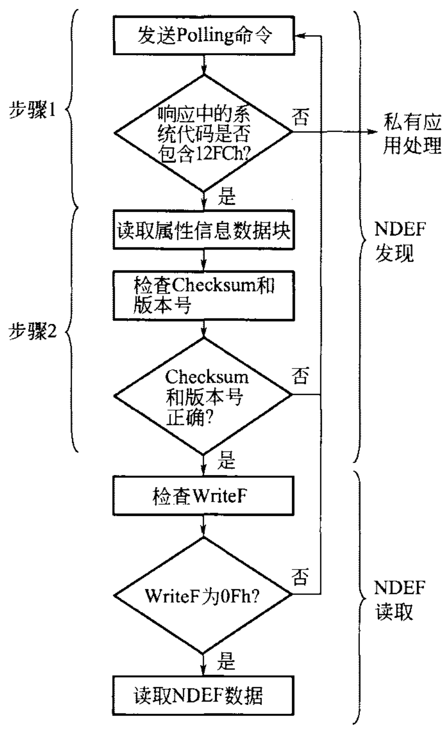


图 6-40 NDEF 数据读操作流程

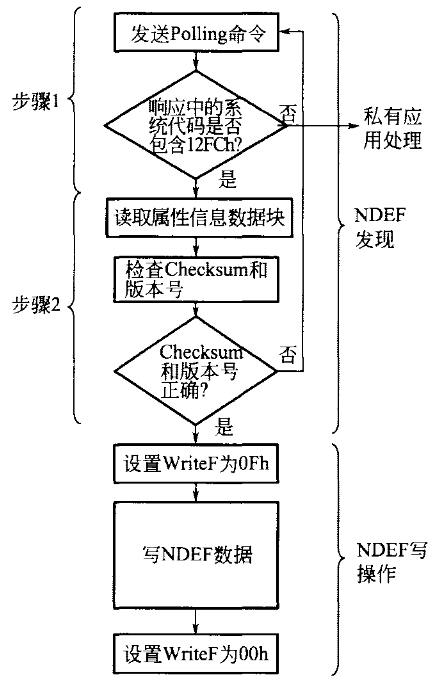


图 6-41 NDEF 数据写操作流程

6.5 类型 4 标签操作规范

6.5.1 存储结构和管理

类型 4 标签采用 ISO 7816-4 定义的文件系统来进行数据存储和管理。ISO 7816 称为集成电路卡（Integrated Circuit Card）规范，也称为接触式智能卡规范。接触式智能卡如常见的电话 IC 卡、电卡、房卡等。手机的 SIM 卡也是接触式智能卡的一种。ISO 7816 一共分为 6 个部分：

① ISO 7816-1：接触式智能卡的物理特性。

② ISO 7816-2：触点位置和卡的尺寸。

③ ISO 7816-3：电信号和传输协议。

④ ISO 7816-4：交换行业间指令。

⑤ ISO 7816-5：应用标识符的编号及登记。

⑥ ISO 7816-6：行业数据元。

ISO 7816 虽然是接触式智能卡的规范，但它实际上是与传输介质无关的应用层协议。在非接触智能卡（即 ISO 14443 规范定义的智能卡）普及之前，接触式智能卡可以说占据了绝大部分市场，而且接触式智能卡的应用非常成熟，因此非接触智能卡沿用了 ISO 7816-4 规范的内容。这也是为什么我们看到，ISO 14443 没有应用相关规范。

ISO 7816-4 包括两部分内容：文件系统和安全功能。文件系统实际上定义了文件存储方式以及访问文件的一系列方法，即 APDU（Application Protocol Data Unit）指令。

ISO 7816-4 的文件系统由 DF（Dedicated File，专用文件）和 EF（Elementary File，基本文件）组成，文件树结构如图 6-42 所示。其中，MF 称为主文件（Master File），是一种特殊的 DF，其实就是处于根位置的 DF。可以将 DF 比作文件目录，MF 为根目录，EF 为文件。EF 中存放数据信息。文件可以通过文件标识符、引用路径或者 DF 的名称寻址。选中文件后，就可以读取文件中的数据内容。

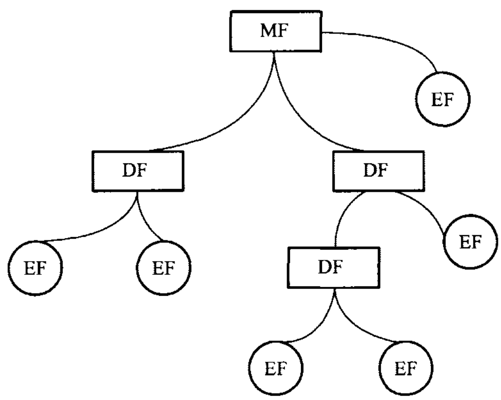


图 6-42 ISO 7816-4 文件树结构

类型 4 标签中至少包含两个 EF 文件：CC 文件和 NDEF 文件。与类型 1/2/3 中的 CC 一样，CC 文件包含标签的存储结构信息，NDEF 文件存放 NDEF 数据。CC 文件的标识符为 E103h。

6.5.2 命令集

类型 4 标签采用 ISO 7816-4 的命令集进行数据读写。命令集如图 6-43 所示。

Select 命令用于选中一个文件或应用。Select 建立了指向智能卡文件系统中一个指定文件的指针，对于该文件的读写操作都需要这个文件指针来定位到文件。ReadBinary 和 UpdateBinary 用于对文件数据的读写操作。这种文件采用的是非结构化二进制形式，因此读写操作需要提供起始字节的偏移量指针和读写长度。

命令/响应	描述
Select	选择应用或文件
ReadBinary	从文件中读数据
UpdateBinary	向文件写数据

图 6-43 类型 4 标签命令/响应集

命令的编码格式如图 6-44 所示。其中，CLA 表示命令类型，可以是 ISO 7816-4 定义的，也可以是私有或厂商定义的。这里取值为 00h，表示是 ISO 7816-4 定义的。INS 为

命令字节，表示具体的指令，如 Select、ReadBinary 等。P1 和 P2 为命令携带的参数，在类型 4 标签操作规范中均设置为 00h，表示没有参数。Lc 为可选，表示数据域的长度。数据域也为可选。Le 称为预期响应数据的长度，也可选。

CLA	INS	P1	P2	Lc(可选)	数据(可选)	Le(可选)

图 6-44 命令编码格式

响应的编码格式如图 6-45 所示。SW1 和 SW2 表示响应的状态信息。

响应数据(可选)	SW1	SW2

图 6-45 响应编码格式

6.5.3 NDEF 的发现及访问

为了发现和访问 NDEF 数据，读写器需要首先读取标签中的 CC 文件。CC 文件包含标签的相关信息，是一种只读的 EF 文件。CC 文件的结构如图 6-46 所示。其中，偏移量是指与 CC 文件起始指针的偏移量。举例来说，CCLEN 的偏移量为 0000h，表示 CCLEN 存放在 CC 文件的起始位置；Version 的偏移量为 0002h，表示 Version 距离 CC 起始位置有 2 字节偏移量。

偏移量(字节)	长度(字节)	名称	描述
0000h	2	CCLEN	表示CC文件的长度
0002h	1	Version	类型4标签操作规范的版本号
0003h	2	MLe	ReadBinary命令一次能够读取数据的最大长度
0005h	2	MLc	UpdateBinary命令一次能够写数据的最大长度
0007h	8	NDEF文件控制TLV	NDEF文件的相关信息
000Fh	-	TLV块	可以包含0个或多个私有TLV块

图 6-46 CC 文件结构

NDEF 文件控制 TLV 用于表示标签中 NDEF 数据的相关信息，采用 TLV 方式存储。其 TLV 结构与类型 1 标签中的定义一致，也包括两种不同长度的 TLV。NDEF 文件控制 TLV 的类型值 T 为 04h，长度 L 为 06h，内容 V 包括 2 字节文件标识符、2 字节最大 NDEF 文件尺寸、1 字节 NDEF 文件读权限和 1 字节 NDEF 写权限。其中，文件标识符用于指示一个 NDEF 文件。

NDEF 消息存放在 EF 文件中，称为 NDEF 文件。读写器通过 NDEF 文件控制 TLV 来确认标签中是否包含 NDEF 文件。NDEF 文件的格式如图 6-47 所示。

读写器通过 NDEF 发现流程来发现类型 4 标签中的 NDEF 消息。发现流程的步骤如下所述。

① 读写器向标签发送 Select 命令来选中 NDEF 标签应用。类型 4 标签符合 ISO 14443 标准，是一种非接触的智能芯片卡，由处理器、ROM、RAM、I/O 接口等组成，在卡上可

偏移量(字节)	长度(字节)	名称	描述
0h	2	NLEN	NDEF消息的长度
2h	×	NDEF消息	NDEF消息内容

图 6-47　NDEF 文件

以运行操作系统，如 JAVA CARD，并且运行各种应用程序。应用程序维护管理相应的数据文件。读写器要访问标签中的数据，首先要选中标签中的某个应用程序。这里所说的 NDEF 标签应用，是维护管理 NDEF 数据的应用程序。Select 命令的格式如图 6-48 所示。其中，CLA 为 00h，表示 ISO 7816-4 定义的命令。INS 为 A4h，表示 Select 命令。P1 为 04h，表示通过名称来选择应用程序。P2 取值为 00h。Lc 取值 07h，表示数据长度为 7 个字节。Le 取值为 00h，表示响应消息中可以包含数据。数据域中为 NDEF 标签应用的文件名。

CLA	INS	P1	P2	Lc	数据	Le
00h	A4h	04h	00h	07h	D2760000850101h	00h

图 6-48　选中 NDEF 标签应用的 Select 命令格式

标签收到 Select 命令后，返回响应消息，其格式如图 6-49 所示。如果标签中包含 NDEF 标签应用（通过 Select 中的文件名匹配），标签将返回 FCI 文件控制信息，其中包含该应用的文件目录。如果标签中没有 NDEF 标签应用，响应消息中将不包含数据。

数据	SW1	SW2	描述
File control information(文件控制信息)	90h	00h	FCI中包含文件的基本结构
-	6Ah	82h	如果未发现NDEF标签应用，则数据域为空

图 6-49　选中 NDEF 标签应用的 Select 命令响应格式

② 如果成功选中 NDEF 标签应用，读写器继续向标签发送 Select 命令来选中 CC 文件。Select 命令格式如图 6-50 所示。其中，P1 取值 00h，表示通过文件标识符来选择文件。数据域中包含 CC 文件的文件标识符。该 Select 命令中不包含 Le，表示不需要响应消息中包含数据。

CLA	INS	P1	P2	Lc	数据
00h	A4h	00h	0Ch	02h	E103h

图 6-50　选中 CC 的 Select 命令格式

数据	SW1	SW2	描述
-	90h	00h	命令成功完成
-	6Ah	82h	CC文件未找到

图 6-51　选中 CC 的 Select 命令响应格式

标签收到 Select 命令后，返回响应消息，其格式如图 6-51 所示。由于 Select 命令中没有包含 Le，因此响应中的数据域不携带任何数据。

③ 成功选中 CC 后，读取 CC 文件并选择 NDEF 文件。读写器向标签发送 ReadBinary 命令来读取 CC 文件，ReadBinary 命令的格式如图 6-52 所示。INS＝B0h，表示 ReadBinary

命令。P1 为偏移量，表示与基准地址（Select 选中的 CC 文件的起始地址）的偏移字节数。举例来说，如果 P1 设置为 00h，而 Le 设置为 0Fh，表示读取 CC 文件中的前 15 个字节。

CLA	INS	P1	P2	Lc	数据	Le
00h	B0h	偏移量		-	-	期望读取的长度

图 6-52 读取 CC 的 ReadBinary 命令格式

标签收到 ReadBinary 命令后，根据 P1 和 Le 返回相应的数据。响应消息格式如图 6-53 所示。

数据	SW1	SW2	描述
读取的数据	90h	00h	返回读取的数据

图 6-53 读取 CC 的 ReadBinary 命令响应格式

④ 读写器解析 CC 文件中的内容，并且读取 NDEF TLV，获取 NDEF 数据的长度及偏移地址。

⑤ 读写器发送 Select 命令选中 NDEF 文件。Select 命令格式如图 6-54 所示。其中，P2 取值 0Ch，表示通过文件标识符来选择 NDEF 文件。Select 命令中不包含 Le，表示响应消息中不包含数据域。

CLA	INS	P1	P2	Lc	数据	Le
00h	A4h	00h	0Ch	02h	文件标识符	-

图 6-54 选择 NDEF 文件的 Select 命令格式

标签收到该 Select 命令后，将返回响应消息。响应消息中不包含数据域。响应消息格式如图 6-55 所示。

数据	SW1	SW2	描述
-	90h	00h	命令成功完成
-	6Ah	82h	NDEF文件未找到

图 6-55 选择 NDEF 文件的 Select 命令响应格式

⑥ NDEF 文件被选中后，读写器就可以对 NDEF 文件进行读写操作了。以写操作为例，UpdateBinary 命令格式如图 6-56 所示。

CLA	INS	P1	P2	Lc	数据	Le
00h	D6h	偏移量		待写入数据的长度	待写入的数据	-

图 6-56 读取 NDEF 文件的 UpdateBinary 命令格式

标签收到 UpdateBinary 命令后，将要写入的数据存放在偏移量指定的位置，并返回响应消息。响应消息如图 6-57 所示。

数据	SW1	SW2	描述
-	90h	00h	命令完成

图 6-57　读取 NDEF 文件的 UpdateBinary 命令的响应格式

6.6　小结

本章介绍了 4 种类型的标签操作规范，包括 4 种类型标签的存储结构和存储管理方式、命令/响应集以及 4 类标签如何进行 NDEF 数据的发现和读写。

第7章 NDEF规范

7.1 NDEF规范概述

NDEF(NFC Data Exchange Format，NFC 数据交换格式）规范[1]定义了 NFC 设备之间或 NFC 设备与标签之间交换消息的封装格式。NDEF 是一个轻量级的二进制消息格式，可以将一个或多个任意类型和大小的应用数据封装进一个 NDEF 消息结构中。应用数据称为 NDEF 消息的负载，用类型、长度和标识符（可选）来描述。类型可以是 URI、MIME 媒体类型，或者是 NFC 特有的类型，如 NFC Forum 定义的一些通用的已知类型，或者是一些组织自己分配的用于特殊目的的类型。长度用于指示 NDEF 消息中负载的长度，用字节数表示。负载的标识符是可选的，一般用于关联多个负载数据，使这些数据之间能够交叉引用。NDEF 消息还可以嵌套。

NDEF 规范定义了 NDEF 的数据结构和构成有效的 NDEF 消息的一系列规则。NDEF 消息由 Record（记录）组成。关于记录，将在第 8 章介绍。NDEF 定义的是数据格式，对于底层的传输协议没有任何依赖性。举例来说，两个 NFC 设备靠近后，通过 LLCP 传输 NDEF 消息。读写器和标签靠近后，NDEF 消息通过标签的相关协议传输。在这两种场景下，NDEF 数据消息都是一样的。

NDEF 提供了一套有效的简单的消息封装机制，可以封装任意类型的数据，如 XML、图片等，而且不同类型的数据可以封装在一个 NDEF 消息中。NDEF 没有重新设计一套数据格式，它实际上是利用已有的数据格式，将它们封装在 NDEF 消息中。

NDEF 规范中没有特别关注安全问题，例如在收到一个 NDEF 消息后，如何保证该 NDEF 消息中不包含安全威胁，这些应当由应用程序来处理。

7.2 NDEF消息格式

一个 NDEF 消息包含了一个或多个 NDEF Record，如图 7-1 所示。第一个 Record 称为 MB Record(Message Begin)，最后一个 Record 称为 ME Record(Message End)。如果 NDEF 消息中只有一个 Record，那么该 Record 既是 MB Record 也是 ME Record。MB 和

---

[1] NFC Data Exchange Format (NDEF) Technical Specification，NDEF 1.0.

ME 是 Record 中的标志位，如果设置为 1，表示该 Record 为 MB 或 ME Record。

NDEF消息						
R_1MB=1	…	R_r	…	R_s	…	R_tME=1

图 7-1 NDEF 消息格式

7.3 Record 格式

7.3.1 Record 结构

Record 实际上就是在 NDEF 消息中携带负载的一个单元，如图 7-2 所示。需要注意的是，Record 没有重新定义一套数据格式，而是将不同类型的负载以相应的 Record 格式封装起来。这些 Record 又可以组成一个 NDEF 消息。

负载包含负载的长度、类型、ID 和负载数据。其中，类型可以是 URI、MIME 媒体类型，或者是 NFC 特有的类型。负载 ID 是可选的，ID 的编码格式采用 URI 规则。通过负载 ID，允许其他负载使用 URI 链接技术引用该负载。

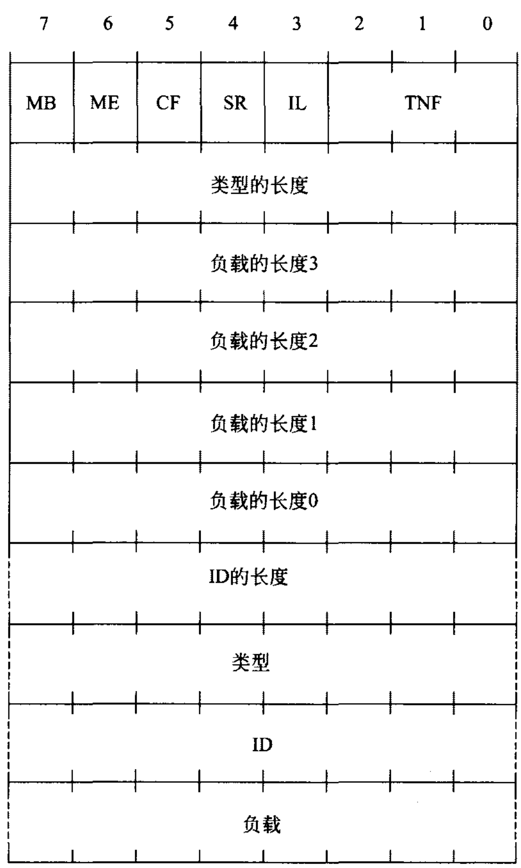


图 7-2 Record 结构

7.3.2 MB

MB 标志位为 1bit。如果设置为 1，表示该 Record 为 NDEF 消息中的第一个 Record。

7.3.3 ME

ME 标志位为 1bit，用于标识 NDEF 消息的最后一个 Record。如果 MB 和 ME 同时设置为 1，表示 NDEF 消息只包含一个 Record。

7.3.4 CF

CF(Chunk Flag) 标志位为 1bit。设置为 1，表示第一个 Record Chunk 或中间的 Record Chunk。

Record Chunk 一般用于将动态产生的长度无法确定的负载数据，或者是非常长的负载数据分段。负载数据被分成多个 Record，每个 Record 称为一个 Record Chunk。这些 Record Chunk 按照顺序依次包含在一个 NDEF 消息中。Chunk 机制实际上主要是考虑负载数据发送方的缓存限制，因此将较长的数据分成多个 Chunk，如图 7-3 所示。

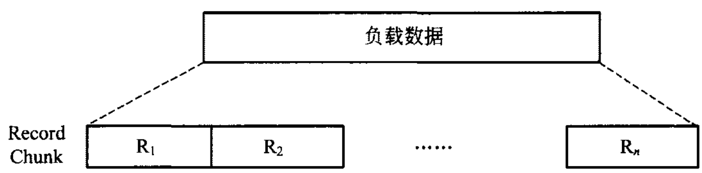


图 7-3 Record Chunk 举例

NDEF 消息包含 0 个或多个 Record Chunk。这些 Record Chunk 的负载类型、ID 必须是一致的。第一个 Record Chunk 的 CF 标志位应当设置为 1。中间 Record 的 CF 标志位也必须设置为 1，并且类型长度和 IL 都必须设置为 0。TNF 设置为 0x06，表示 TNF 与前一个 Record 一致。Record Chunk 的 CF 标志位设置为 0，表示这是最后一个 Record Chunk。同样，类型长度和 IL 都必须设置为 0。TNF 设置为 0x06，表示 TNF 与前一个 Record 一致。

所有的 Record Chunk 都必选完全封装在一个 NDEF 消息中，不允许分布在多个 NDEF 消息中。ME 标志位不能在第一个和中间 Record Chunk 中被设置为 1。

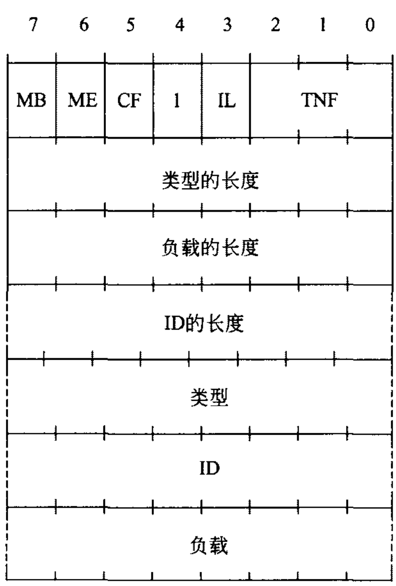


图 7-4 SR 设置为 1 的 Record 结构

7.3.5 SR

SR(Short Record) 标志位如果设置为 1，则负载长度用一个字节表示。SR 主要用于长度为 0～255 字节的 Record。图 7-4 所示为 SR 标志位设置为 1 的 Record 结构。

7.3.6 IL

IL(ID Length) 标志位如果设置为 1，表示 ID 长度域将包含在 Recrod 中。如果设置为 0，表示 ID 长度

以及 ID 都不会包含在 Record 中。因为 ID 是可选的，所以在不需要 ID 的 Record 中，可以将 IL 设置为 0。

7.3.7　TNF

TNF(Type Name Format) 用于指示 Record 类型域的类别。TNF 取值如表 7-1 所示。

表 7-1　TNF 取值

TNF	值	TNF	值
空	0x00	NFC Forum 扩展类型	0x04
NFC Forum 已知类型(RTD 规范)	0x01	未知	0x05
媒体类型(RF2406)	0x02	Unchanged(未改变)	0x06
URI(RF3986)	0x03	预留	0x07

TNF 设置为 0x00h，表示 Record 中没有类型域或负载域，一般用于空的 Record。例如，当没有负载数据时，用一个空的 Record 表示 NDEF 消息结束。在这种情况下，Record 中的类型长度、ID 长度和负载长度将被设置为 0，类型、长度和负载被忽略。

TNF 设置为 0x01h，表示 Record 中的类型域为 NFC Forum RTD 规范定义的类型，将在第 8 章详细介绍，如 URI、文本等。

TNF 设置为 0x02h，表示 Record 中的类型域符合 RFC 2406 定义的媒体类型。

TNF 设置为 0x03h，表示 Record 中的类型域符合 RFC 3986 定义的 URI 类型。

TNF 设置为 0x04h，表示 Record 中的类型域符合 NFC Forum 定义的扩展类型。

TNF 设置为 0x05h，表示 Record 中的类型域为未知类型，此时类型长度域必须设置为 0，类型域被忽略。

TNF 设置为 0x06h，表示该 Record 为中间和最后的 Record Chunk。该值只能用于 Record Chunk，类型长度域必须设置为 0，类型被忽略。

7.3.8　类型长度

类型长度域的值表示类型域的长度。

7.3.9　ID 长度

ID 长度域的值表示负载 ID 的长度。当 IL 设置为 1 时，Record 才包含该域。如果 IL 设置为 0，Record 将不包含该域。

7.3.10　负载长度

负载长度域的值表示负载的长度。负载长度域的长度由 SR 确定。SR 设置为 1 时，表示负载长度域为 1 字节；设置为 0 时，表示负载长度域为 4 字节。

7.3.11　类型

类型域描述了负载的类型。类型域需要符合 TNF 规定的类型结构、编码以及格式。如 TNF 取值为 0x01，表示 NFC Forum 已知类型，类型域的取值可以是 URI、文本、智能海报等。

7.3.12 ID

ID 采用 URI 编码格式，主要用于 Record 之间互相引用。举例来说，一封电子邮件可能包含邮件正文、邮件附件的图片、邮件的收/发件人，这些数据被放置在不同的 Record 中，而且这些 Record 之间有一定的关联和引用关系。

在 Record Chunk 中，中间和最后的 Record Chunk 不包含 ID 域。

7.3.13 负载

负载域包含用户数据。

7.4 小结

NDEF 规范定义了 NDEF 的数据结构和构成有效 NDEF 消息的一系列规则。NDEF 提供了一套有效的简单的消息封装机制，可以封装任意类型的数据。本章首先介绍了 NDEF 规范的应用领域，NDEF 是 NFC 设备之间或 NFC 设备与标签之间交换消息的封装格式；然后描述了 NDEF 消息格式的总体架构，NDEF 由一个或多个 Record 组成；最后详细介绍了 Record 的组成以及各个域的含义和取值。

第8章 RTD规范

8.1 RTD规范概述

第 7 章介绍过，NDEF 支持不同类型的负载数据，如标准的 MIME 和 URI，以及 NFC Forum 定义的已知类型和扩展类型等。RTD(Record Type Definition) 规范[1]将详细介绍 NFC Forum 定义的已知类型（NFC Forum Well Known Types）和 NFC 扩展类型（NFC External Types)。

在 Record 中，如果 TNF 取值为 0x01h，表示该 Record 的负载数据类型为 NFC Forum 已知类型。如果 TNF 取值为 0x04h，表示该 Record 的负载数据类型为扩展类型。

NFC Forum 目前定义了 4 种类型的 RTD 规范：文本 RTD、URI RTD、智能海报 RTD 和签名 RTD。除此之外，第三方也可以根据 RTD 规则定义自己的 Record。

8.1.1 NFC Forum 已知类型

NFC Forum 已知类型采用 URN(Uniform Resource Name) 命名规则，即以“urn:”开头，后面跟随着命名空间标识符（NID）“nfc”和已知类型“wkt”。举例来说，NFC Forum 已知类型的类型名称为“urn: nfc: wkt: abc”。其中，“urn: nfc: wkt:”为前缀，“abc”为实际的类型名称。一般情况下，Record 的类型名称省略前缀。例如 NFC Forum 定义的文本 RTD，在 Record 中的类型为“T”，但实际上完整的类型名称为“urn: nfc: wkt: T”。

在类型的命名规则上，如果是全局类型（Global Type），NFC Forum 要求必须以大写字母开始，如“U”、“Cfq”、“Trip-to-Texas”等。NFC Forum 定义的 4 个 RTD 规范的 Record 类型均为全局类型，文本 Record 为“T”，URI Record 为“U”，智能海报 Record 为“Sp”，签名 Record 为“Sig”。

如果是本地类型（Local Type），NFC Forum 要求必须以小写字母或数字开头，如“0”、“foo”、“u”等。本地类型一般由第三方或开发者自己定义，作用域仅在本

[1] NFC Record Type Definition（RTD）Technical Specification，RTD 1.0.

Record 中。

8.1.2 NFC Forum 扩展类型

NFC Forum 扩展类型同样采用 URN 命名规则，以“urn：nfc：ext：”作为前缀。例如“urn：nfc：ext：example.com：f”，其中“example.com：f”为类型名称。在 NFC Forum 扩展类型中，要求类型名称必须包含发布组织的域名，域名之后为冒号，然后跟随实际的类型名称。例如，“example.com”为该扩展类型的发布组织的域名，“f”为实际的类型名称。

8.2 文本 RTD 规范

文本 Record 规范[1]定义了文本数据的 Record 类型，属于 NFC Forum 已知类型，一般用于标签中自由格式的文本描述。

文本 RTD 主要用作一些对象的元数据，例如为 URL 做一些文字上的说明。因此需要提供一种轻量级的文本描述，它并不是要替换目前的文本标准，而是定义一个子集用于对空间有限制的应用场景，如标签应用。

文本 Record 的 TNF 设置为 0x00h，类型域为“T”，正如前面提到的，实际文本 Record 完整的类型为“urn：nfc：wkt：T”。文本 Record 内容的编码格式如图 8-1 所示。语言代码一般为 2 个或 4 个字符。在图 8-1 中，“fi”表示芬兰语，“jp”表示日本语，“fr”表示法语。

偏移量（字节）	长度（字节）	内容
0	1	b7=0，表示文本编码为UTF-8；b7=1，表示文本编码为UTF-16。b6为预留位，设置为0；b5～b0为IANA语言代码的长度
1	<n>	ISO/IANA语言代码，如“fi”、“jp”、“en-US”、“fr-CA”等，编码格式采用US-ASCII
n+1	<m>	实际的文本内容，可以用UTF-8或UTF-16编码

图 8-1 文本 Record 内容编码格式

图 8-2 以英文“hello，world!”为例，说明文本 Record 的结构，NDEF 消息中只包含一个文本 Record。

[1] Text Record Type Definition Technical Specification，RTD-Text 1.0.

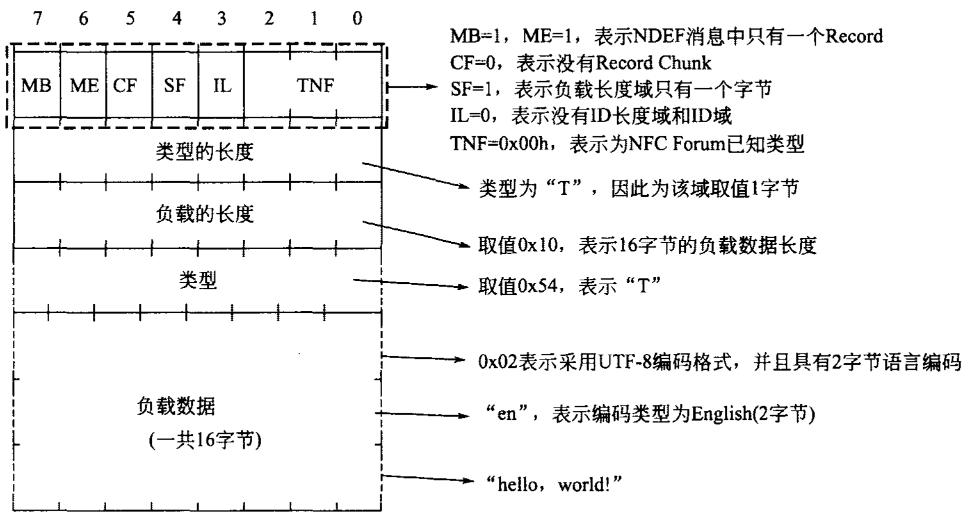


图 8-2　“hello world”文本 Record 举例

8.3　URI RTD 规范

URI RTD 规范[1]定义了使用 URI 类型（包括 URN 或 URL）的负载数据的编码格式。URI Record 的 TNF 设置为 0x00h，类型域为“U”。

如图 8-3 所示，URI Record 负载数据包括两个部分，第一部分为 1 字节的标识符代码，第二部分为 URI 域。

名称	偏移量	长度	取值	描述
标识符代码	0	1 byte	URI标识符代码	URI域前缀
URI域	1	N	UTF-8字符串	余下的URI部分，或者完整的URI(当标识符代码为0x00h时)

图 8-3　URI Record 负载数据格式

为了缩短 URI，Record 负载数据的第一个字节描述了 URI 的协议域，如“http://www.”、“ftp://”等。图 8-4 描述了 URI 标识符代码的格式。可以通过 URI 标识符代码将 URI 域中的数据分发给相应的应用程序。例如，URI 标识符代码为 0x01，可以启动 http 服务来解析。URI 标识符代码和 URI 域中的内容组成一个完整的 URI。举例来说，如果 URI 标识符代码为 0x01，URI 域中的内容为“nfc-forum.org”，那么完整的 URI 为“http://www.nfc-forum.org”。如果标识符代码取值为 0x00，表明 URI 域中是一个完整的 URI。URI 域采用 UTF-8 编码格式。

举例来说，将 URL“http://www.nfc.com”存放在 NDEF 消息中，NDEF 消息将使用 URI Record 来封装这个 URL，如图 8-5 所示。

将电话号码“+35891234567”存放在 NDEF 消息中。NDEF 消息将使用 URI Record 封装电话号码，如图 8-6 所示。

[1] URI Record Type Definition Technical Specification，RTD-URI 1.0.

将私有 URI 数据 "mms：// example. com/download. wmv" 存放在 NDEF 消息中，NDEF 消息将使用 URI Record 封装该 URI，如图 8-7 所示。URI 标识符代码设置为 0x00，表示 URI 域包含完整的 URI。

十进制数	十六进制数	协　议
0	0x00	N/A.No prepending is done.and the URI field contains the unabridged URL
1	0x01	http：//www.
2	0x02	https：//www.
3	0x03	http：//
4	0x04	https：//
5	0x05	tel：
6	0x06	mailto：
7	0x07	ftp：//anonymous：anonymous@
8	0x08	ftp：//ftp.
9	0x09	ftps：//
10	0x0A	sftp：//
11	0x0B	smb：//
12	0x0C	nfs：//
13	0x0D	ftp：//
14	0x0E	dav：//
15	0x0F	news：
16	0x10	telnet：//
17	0x11	imap：
18	0x12	rtsp：//
19	0x13	urn：
20	0x14	pop：
21	0x15	sip：
22	0x16	sips：
23	0x17	tftp：
24	0x18	btspp：//
25	0x19	btl2cap：//
26	0x1A	btgoep：//
27	0x1B	tcpobex：//
28	0x1C	irdaobex：//
29	0x1D	file：//
30	0x1E	urn：epc：id：
31	0x1F	urn：epc：tag：
32	0x20	urn：epc：pat：
33	0x21	urn：epc：raw：
34	0x22	urn：epc：
35	0x23	urn：nfc：
36...255	0x24..0xFF	RFU

图 8-4　URI 标识符代码

偏移量	内容	描述
0	0xD1	SR=1，MB=ME=1，TNF=0x01，CF=0,IL=0，SF=1
1	0x01	类型的长度为1字节
2	0x08	负载数据长度为8字节
3	0x55	类型为“U”，表示URI
4	0x01	URI标识符为“http：//www.”
5	0x6e 0x66 0x63 0x2e 0x63 0x6f 0x6d	UIF-8的编码：nfc.com

图 8-5　URI Record 实例-1

偏移量	内容	描述
0	0xD1	SR=1，MB=ME=1，TNF=0x01，CF=0，IL=0，SF=1
1	0x01	类型的长度为1字节
2	0x0D	负载数据长度为13字节
3	0x55	类型为“U”，表示URI
4	0x05	URI标识符为“tel：”
5	0x2b 0x33 0x35 0x38 0x39 0x31 0x32 0x33 0x34 0x35 0x36 0x37	UTF-8的编码：+35891234567

图 8-6　URI Record 实例-2

偏移量	内容	描述
0	0xD1	SR=1，MB=ME=1，TNF=0x01，CF=0，IL=0，SF=1
1	0x01	类型的长度为1字节
2	0x1F	负载数据长度为31字节
3	0x55	类型为“U”，表示URI
4	0x00	无前缀
5	0x6d 0x6d 0x73 0x3a 0x2f 0x2f 0x65 0x78 0x61 0x6d 0x70 0x6c 0x65 0x2e 0x63 0x6f 0x6d 0x2f 0x64 0x6f 0x77 0x6e 0x6c 0x6f 0x61 0x64 0x2e 0x77 0x6d 0x76	字符串：“mms：//example.com/download.wmv”

图 8-7　URI Record 实例-3

8.4 智能海报 RTD 规范

智能海报 RTD（The Smart Poster RTD）规范[1]定义如何将 URL、SMS（Short Messages，短信）或电话号码等信息存放在标签中，并且如何将这些内容在设备与设备、设备与标签之间传递。

在智能海报的应用场景中，用户通过 NFC 手机读取海报内容。这些内容可能包括一个 URL，对该 URL 的文本描述，也可能有一个电话号码。当用户手机获取这些内容后，将自动启动浏览器连接到该 URL 的网页。在启动过程中，向用户展示该 URL 的一些说明，还可能启动短信应用程序向那个电话号码发送一条短信。

可以看出，智能海报应用实际上融合了多种类型的数据，如 URI、文本、动作（如触发某个应用程序）、图片等等。因此 NFC Forum 定义了智能海报 RTD 规范，以便更好地将不同数据类型整合在一个 Record 中。

智能海报 Record 的 TNF 为 NFC Forum 已知类型，负载数据的类型为“Sp”。负载数据由 NDEF 消息组成，即负载数据的内容包括几个 Record。智能海报 Record 包含 0 个、1 个或多个下列 Record：

① 标题 Record(The Title Record)：该 Record 是一个文本 Record，用于描述智能海报的标题。标题 Record 是可选的。

② URI Record：这是智能海报的核心 Record，其他 Record 都是该 Record 的元数据。一个智能海报中只能包含一条 URI Record。

③ 动作 Record(The Action Record)：该 Record 是可选的，它描述了一个动作，如指示手机将 URI 存为标签或者打开浏览器等。动作 Record 是一个本地类型的 Record，在 8.1.1 节介绍过。动作 Record 中的负载数据类型为“act”。因为是本地类型，因此其作用域只局限在该智能海报 Record 中。动作 Record 的负载数据只有 1 字节，取值及含义如图 8-8 所示。如果智能海报 Record 包含动作 Record，它只是作为一个建议，对于开发者来说可以忽略。

值	动作
0	执行动作(发送短信、启动浏览器、拨号)
1	保存(保存短信至收件箱，保存URI至标签，保存电话号码至联系人)
2	打开编辑(打开短信并编辑，打开URI并编辑，打开电话号码并编辑)
3..FF	预留

图 8-8 动作 Record

④ 图标 Record(The Icon Record)：智能海报 Record 可以通过包含多个 MIME 类型的 Record 给用户展示一个图标。图标 Record 是可选的。

⑤ 容量 Record(The Size Record)：如果 URI 引用了一个外部实体，那么容量 Record 可以指示这个外部实体的大小。容量 Record 对于读写器来说非常有用。读写器通过该 Record 中的内容判断是否有能力处理外部实体。容量 Record 是可选的。容量 Record 的负载数据包含 4 字节，如图 8-9 所示。目前只支持 URL 引用的外部实体，如 http：//、ftp：//等。容量 Record 的负载数据类型为“s”，为本地类型。举例来说，如果 Byte0 为 0x12，Byte1 为 0x34，Byte2 为 0x56，Byte3 为 0x78，那么引用的实体对象大小为

[1] URI Record Type Definition Technical Specification，RTD-URI 1.0.

0x12345678 字节。

0	1	2	3
Byte 0	Byte 1	Byte 2	Byte 3

图 8-9 容量 Record

⑥ 类型 Record(The Type Record)：如果 URI 引用了一个外部实体，类型 Record 用于说明该外部实体的类型。举例来说，智能海报可能提供一个电影片段作为宣传，那么类型 Record 会告诉应用该外部实体的类型为 MIME 的视频。类型 Record 负载数据类型为本地类型“t”。

除此之外，智能海报还包含其他 Record，如 vCard 等。在智能海报 Record 中没有指定这些 NDEF Record 的顺序。

为了更好地理解智能海报 Record，下面举两个例子来说明智能海报 Record 的格式。第一个例子是智能海报中包含一个 URI “http：//www. nfc-forum. org”，如图 8-10 所示。从图中可以看出，智能海报 Record 中包含了一个 URI 的 Record，用于存放“http：//www. nfc-forum. org”。

偏移量	内容	长度	描述
0	0xD1	1	SR=1, MB=ME=1, TNF=0x01, CF=0, IL=0, SF=1
1	0x02	1	类型的长度为2字节(sp)
2	0x12	1	负载数据长度为18字节
3	“Sp”	2	类型为“Sp”，表示智能海报
5	0xD1	1	SR=1, MB=ME=1, TNF=0x01, CF=0, IL=0, SF=1
6	0x01	1	类型的长度为1字节(U)
7	0x0E	1	负载数据长度为14字节
8	“U”	1	类型为“U”，表示URI
9	0x01	1	URI标识符为“http：//www.”
10	“nfc-forum.org”	13	URI域

图 8-10 智能海报 Record 实例-1

第二个例子中，智能海报 Record 包含 4 个 Record，如图 8-11 所示。第一个 URI Record 用于存放“http：//www. nfc-forum. org”。第二个 Record 是动作 Record，用于启动浏览器。第三个 Record 为标题 Record，存放英语字符串“Hello World”。最后的 Record 也是标题 Record，用于存放芬兰语“Morjens，maailma”。这个例子中，当用户手机靠近智能海报并获取到包含智能海报 Record 的 NDEF 消息后，如果手机语言设置为芬兰语，那么第四个标题 Record 将显示；如果手机不是芬兰语的，将显示第三个标题 Record。同时浏览器自动启动，打开 NFC Forum 主页。

偏移量	内容	长度	说明
0	0xD1	1	NDEF header TNF=0x01 (Well Known Type).SR=1，MB=1, ME=1
1	0x02	1	Record name length(2 bytes)
2	0x49	1	Length of the Smart Poster data(73 bytes)
3	"Sp"	2	The record name
5	0x81	1	NDEF header.TNF=0x01, SR=0, MB=1, ME=0
6	0x01	1	Record name length(1 byte)
7	0x00，0x00，0x00，0x0E	4	The length of the URI payload (14 bytes)(long format)
11	"U"	1	Record type："U"
12	0x01	1	Abbreviation："http://www."
13	"nfc-forum.org"	13	The URI itself.
26	0x11	1	NDEF record header(SR=1, TNF=0x01)
27	0x03	1	The length of the record name
28	0x01	1	The length of the "act" payload.
29	"act"	3	Record type： "act"
32	0x00	1	Action=Launch browser
33	0x11	1	NDEF：record header(SR=1, TNF=0x01)
34	0x01	1	Length of the record name
35	0x12	1	Length of the record payload (18 bytes)
36	"T"	1	Record type："T" (=Text)
37	0x05	1	Status byte for the Text(UTF-8，five-byte code)
38	"en-US"	5	ISO Language code：US-English
43	"Hello, world"	12	The text："Hello world"，encoded in UTF-8.
55	0x51	1	NDEF record header(SR=1, TNF=0x01, ME=1)
56	0x01	1	Record name length
57	0x13	1	Length of the Text payload (19 bytes)
58	"T"	1	The name of the Text record ("T")
59	0x02	1	Status byte：UTF-8, two-byte language code
60	"fi"	2	ISO two-character language code：Finnish
62	"Morjens, maailma"	16	The text "Morjens，maailma" encoded in UTF-8

图 8-11 智能海报 Record 实例-2

8.5　签名 RTD 规范

对 NDEF 数据进行数字签名是一种值得信任的方法，用户通过这种方式来验证 NDEF 数据的可靠性和完整性。

NFC Forum 制定了签名 RTD 规范[1]。该规范定义了对一个或多个 NDEF Record 签名的格式，并且提供了用于创建签名的算法和证书列表。签名 Record 包含一个数字签名，用于验证 NDEF 消息中一个或多个 Record 内容的完整性和可靠性。

签名 Record 的 TNF 为 NFC Forum 已知类型，类型域为 "Sig"。负载数据包括三个部分：版本号、签名和证书链，如图 8-12 所示。

版本号长度为 1 字节，如图 8-13 所示。目前该签名 RTD 规范的版本为 1.0，因此版本号设置为 0x01。

签名Record		
版本	签名	证书链

图 8-12　签名 Record 的负载数据

7	6	5	4	3	2	1	0
版本							

图 8-13　签名 Record 的版本号

签名域包含一个实际签名，或者是一个能够找到签名所在位置的引用。签名域包含 4 个子域，如图 8-14 所示。

签名域			
URI_Present	签名类型	签名/URI长度	签名/URI
1bit	7bit	16bit	N字节(byte)

图 8-14　签名 Record 的签名域

URI _ Present 标志位为 1bit，用于指示签名域中包含的是签名还是签名的引用位置。图 8-15所示为签名类型，列出了各种签名算法。如果设置为 0x00，表示没有签名。

如果 URI _ Present=0，并且签名类型=0，该签名 Record 将不会对从 NDEF 消息的开始到该签名 Record 之间的 Record 进行验证。签名/URI 长度子域和签名/URI 子域将不会出现在签名域中，证书链域也不会出现在签名 Record 中。在这种情况下，签名 Record 作为一个起始标志，用于指示后面的 Record 将被签名。

如果 URI _ Present=0，并且签名类型不等于 0，签名/URI 域应当包含实际签名，并且签名/URL 长度用于表示签名的长度。

[1] Signature Record Type Definition Technical Specification，SIGNATURE 1.0.

如果 URI _ Present=1，签名类型不等于 0，签名/URI 域将包含一个 URI，表示该签名所在的位置。

如果 URI _ Present 设置为 1，签名类型将不能被设置为 0。

十六进制	签名类型
0x00	没有签名
0x01	RSASSA-PSS SHA-1 [PKCS_1]
0x02	RSASSA-PKCS1-v1_5(有SHA-1) [PKCS_1]
0x03	DSA [DSA]
0x04	ECDSA-P-192 SHA-1 [ECDSA]
0x05～0x7f	预留

图 8-15 签名 Record 的签名类型

证书链域包含一些必选和可选的子域，如图 8-16 所示。URI _ Present 标识符为 1bit，表示证书链域中包含的是证书还是证书的引用。

证书链域						
URI_Present	Cert_Format	Nbr_of_Certs	Cert_Store			Cert_URI
1bit	3bit	4bit	证书0	…	证书*n*	URI域

图 8-16 证书链域

如果 URI _ Present 设置为 1，证书链域中将包含 Cert _ URI 子域。如果 URI _ Present 设置为 0，证书链域中将不包含 Cert _ URI 子域。

Cert _ Format 为 3bit，用于表示签名 Record 中采用的证书格式，如图 8-17 所示。

十六进制	证书格式
0x00	X.509 [X_509]
0x01	X9.68 [X9_68]
0x02～0x07	预留

图 8-17 签名 Record 的证书格式

Nbr _ of _ Certs 为 4bit，表示 Cert _ Store 域中证书的个数，一共可以包含 16 个证书。如果 Nbr _ of _ Certs 大于 0，Cert _ Store 中的第一个证书应当是签名者的证书，后面跟随 0 个或多个证书。Cert _ Store 不应该包含最高层的证书，如根证书。

如果证书链域中包含 Cert _ URI，Cert _ URI 应当是证书链中下一个证书的引用。

Cert _ Store 中证书的格式如图 8-18 所示。长度域为 16bit，用于指示证书长度；数值域则包含证书。

证书	
长度，N	值
16bit	N字节

图 8-18　签名 Record 的证书

URI 域如图 8-19 所示。长度域为 16bit，用于指示 URI 的长度；数值域表示 URI。

URI	
长度，N	值
16bit	N字节

图 8-19　签名 Record 的 URI

签名 Record 的作用域可以是从 NDEF 消息的第一个 Record 到该签名 Record 之间所有的 Record，包括第一个 Record；或者是从上一个签名 Record 之后的第一个 Record 到该签名 Record 之间所有的 Record。签名 Record 不会对自己签名。签名 Record 将会对 Record 中的类型域、ID 域（如果包含）和负载域签名；而对于 NDEF 的第一个字节（如 TNF、SR 等）、类型长度、负载长度、ID 长度，将不做签名。

签名 Record 的作用域如图 8-20 所示。空的签名 Record 是指 URI _ Present＝0，并且签名类型＝0 的签名 Record。该签名 Record 中不包含签名/URI 长度子域和签名/URI 子域，只是作为一个起始标志，指示后面的 Record3 将被签名。签名 Record2 对 NDEF Record3 签名。

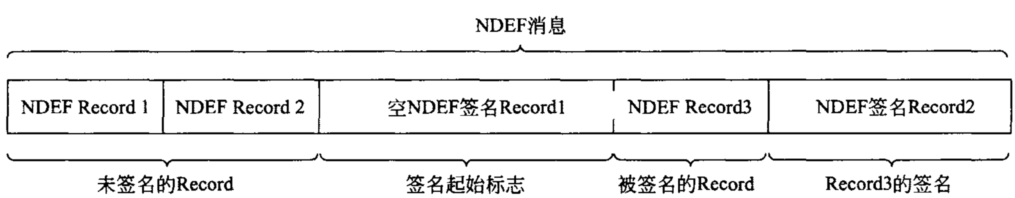


图 8-20　空签名 Record

图 8-21 描述了对智能海报 Record 签名的示例。智能海报 Record 包含的第一个 Record 为 URI Record，第二个 Record 为签名 Record。签名 Record 对 URI Record 签名。

图 8-22 描述了对智能海报 Record 签名的示例。智能海报 Record 包含 4 个 Record，第一个为文本 Record；第二个为签名 Record，该签名 Record 对文本 Record 签名；第三个 Record 为 URI Record；第四个 Record 为签名 Record，该签名 Record 对 URI Record 签名。

图 8-23 描述了 NDEF 消息中包含 3 个 Record 的示例。第一个 Record 为文本 Record；第二个 Record 为 URI Record；最后一个为签名 Record，该签名 Record 对文本 Record 和 URI Record 签名。

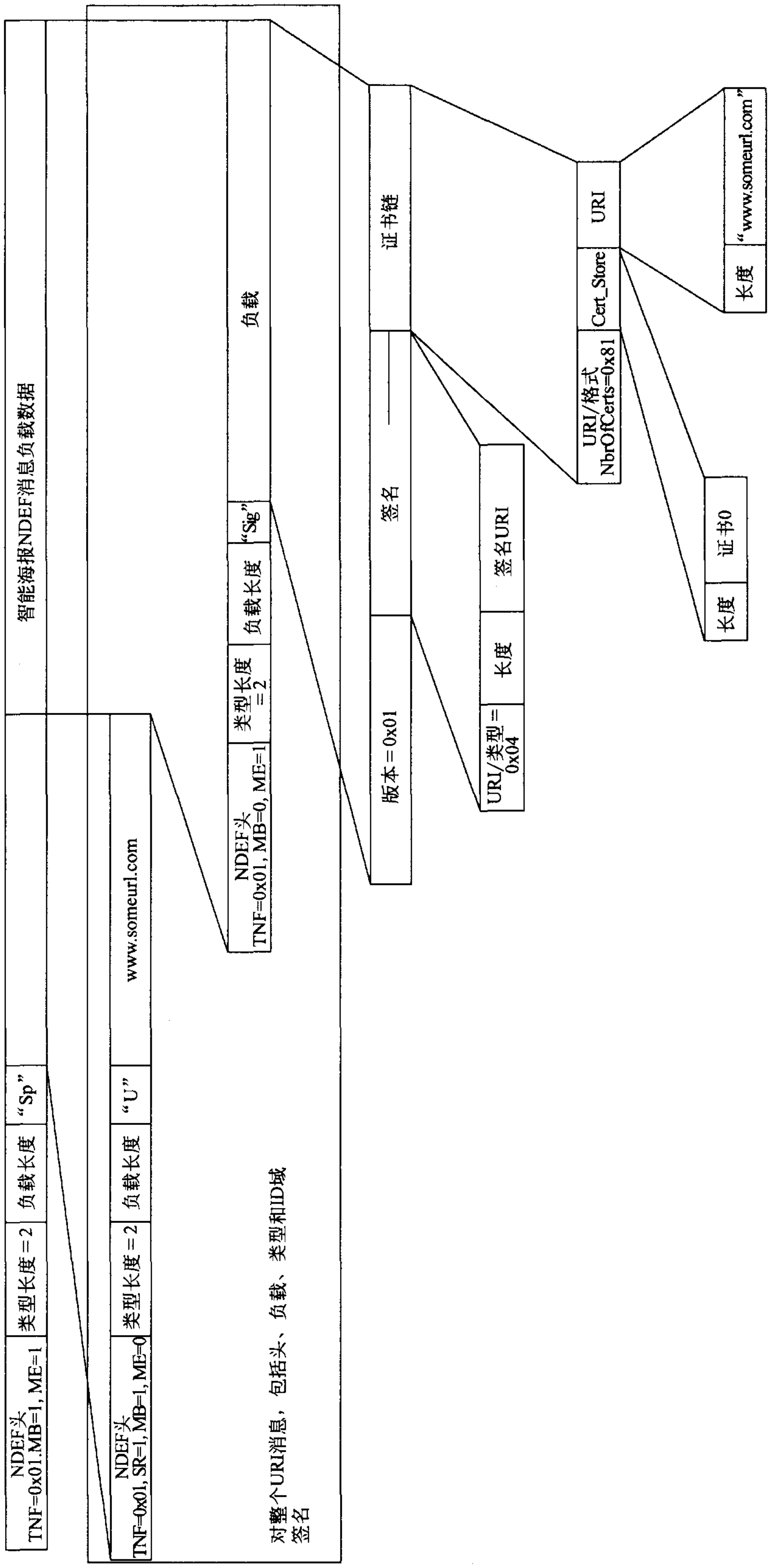


图8-21 签名Record示例-1

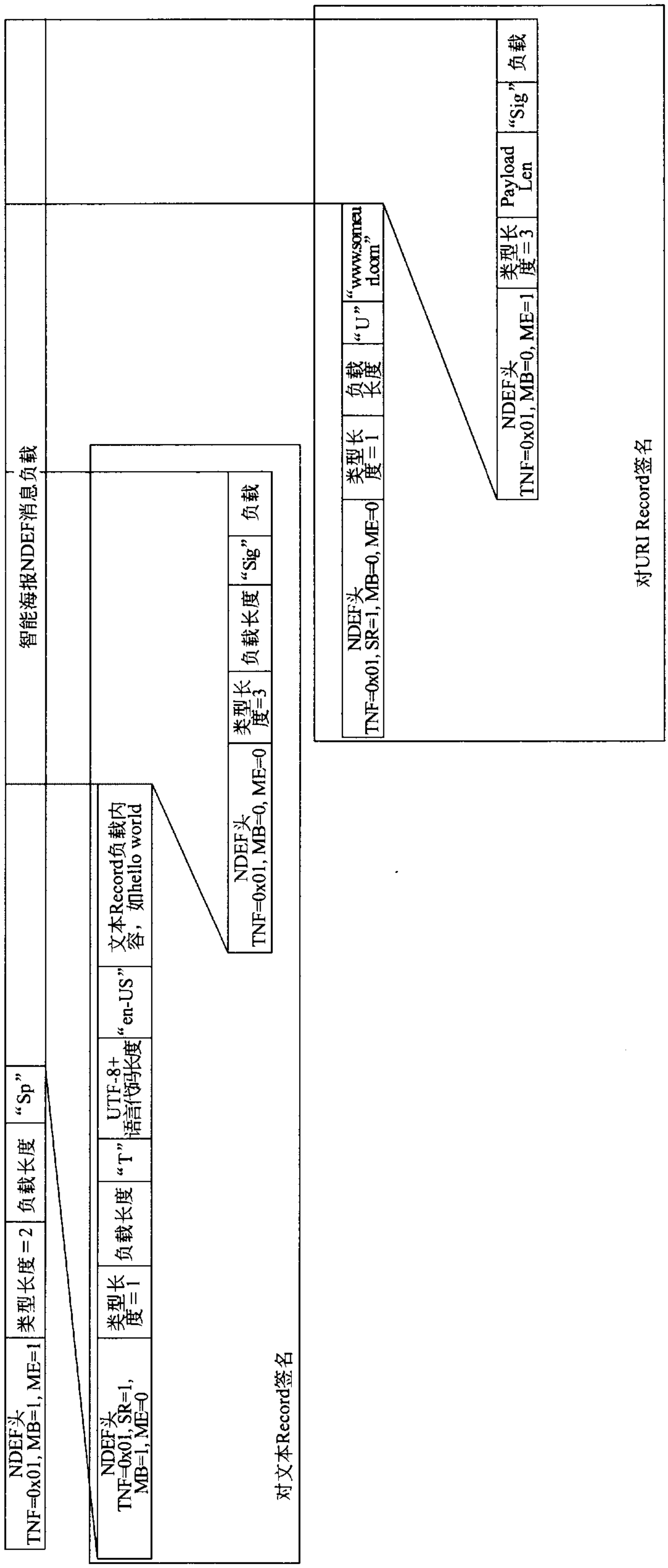


图8-22 签名Record示例-2

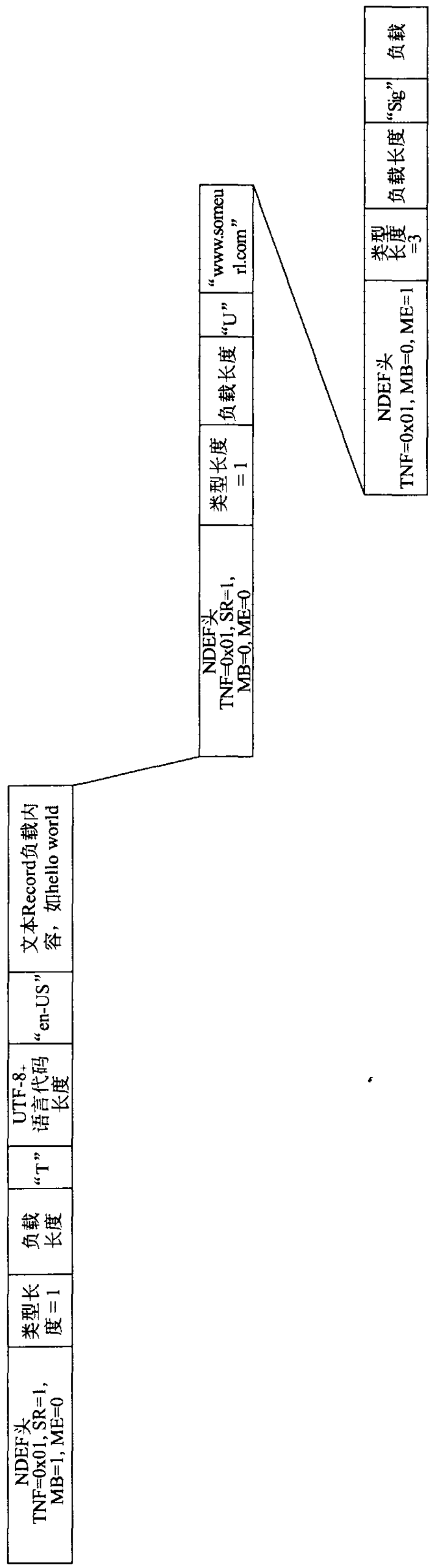


图8-23 签名Record示例-3

8.6 小结

NDEF 消息由 Record 组成。本章介绍了 RTD 规范，包括文本 RTD 规范、URI RTD 规范、智能海报 RTD 规范和签名 RTD 规范。针对每一个规范详细阐述，包括文本/URI/智能海报/签名 Record 的数据格式，每一个域的含义、应用场景，并给出示例，以便读者掌握 RTD 规范。

Retro Workshop
果粉工作室

第9章 SNEP规范

9.1 SNEP规范概述

SNEP(Simple NDEF Exchange Protocol)❶ 是一个应用层协议，用于点对点通信中两个NFC 设备交换 NDEF 数据。SNEP 采用请求/响应的协议架构，如图 9-1 所示。

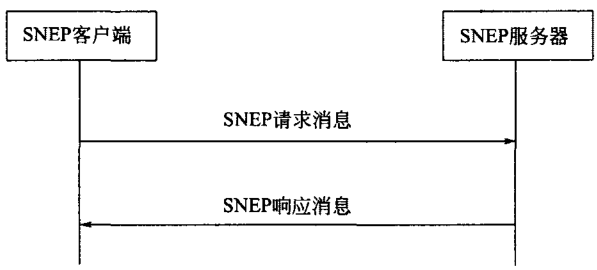


图 9-1　SNEP 协议架构

SNEP 客户端向 SNEP 服务器发送请求消息。请求消息中包含协议版本、请求代码、信息域的长度和信息域。SNEP 根据请求消息中的请求代码进行响应。响应消息中包含协议版本、用于指示成功或失败的状态码、信息域的长度和信息域。SNEP 消息结构如图 9-2 所示。SNEP 的目的是在 NFC 设备之间交换 NDEF 消息，NDEF 消息承载在请求和响应消息的信息域中。SNEP 支持的最大 NDEF 消息为 $2^{32}-1$ 个字节。

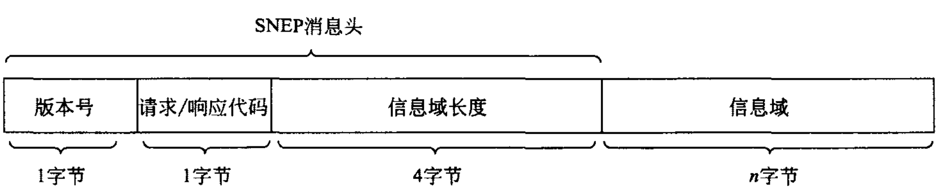


图 9-2　SNEP 消息结构

❶ Simple NDEF Exchange Protocol Technical Specification，SNEP 1.0.

9.2 SNEP 消息

9.2.1 SNEP 消息传输

传输 SNEP 消息需要可靠的传输协议。在 NFC Forum 协议架构中，SNEP 基于 LLCP 协议，LLCP 提供了面向连接的传输服务来保证 SNEP 消息可靠传输。

LLCP 为 SNEP 预留的服务接入点为 4，服务名称为“urn：nfc：sn：snep”。SNEP 客户端向本地 LLC 发出请求，请求本地 LLC 建立与远端设备 LLC 的连接，该连接的服务接入点为 4。SNEP 服务器将等待 LLC 的连接请求。LLC 连接建立成功后，SNEP 客户端和 SNEP 服务就可以通信了。

9.2.2 SNEP 请求消息

SNEP 客户端向 SNEP 服务器发送请求消息来调用服务器上的某个方法。SNEP 请求消息如图 9-3 所示。信息域长度表示整个信息域的字节数，信息域中存放 NDEF 消息。

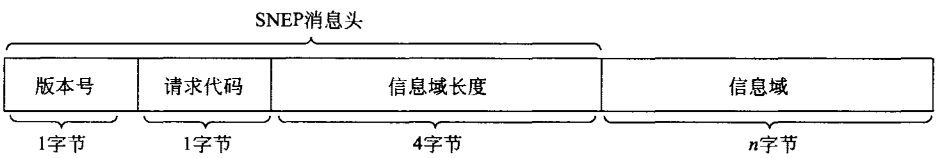


图 9-3 SNEP 请求消息

版本号用于指示 SNEP 的协议版本号，如图 9-4 所示。b7～b4 表示主版本号，b3～b0 表示次版本号。

请求代码用于 SNEP 客户端请求 SNEP 服务器执行的操作。请求代码域的取值如图 9-5 所示。

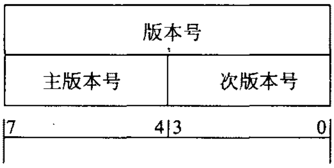


图 9-4 SNEP 协议版本号

请求代码	名称	描述
00h	Continue	请求SNEP服务器发送剩余的SNEP分段消息
01h	Get	请求SNEP服务器返回一个NDEF消息
02h	Put	请求SNEP服务器接受一个NDEF消息
03h-7Eh		预留
7Fh	Reject	请求SNEP服务器不要发送剩余的SNEP分段消息
80h-FFh		预留

图 9-5 SNEP 请求代码

SNEP 消息可以携带最大 $2^{32}-1$ 个字节的 NDEF 消息。如果 SNEP 请求或响应消息超过 LLC 规定的最大长度，SNEP 消息将分段传输。分段将在 9.2.4 节介绍。Continue 用于 SNEP 客户端请求 SNEP 服务器发送剩余的 SNEP 分段响应消息。

Get 用于 SNEP 客户端请求 SNEP 服务器返回一个 NDEF 消息，信息域中包含 4 字节可接收长度域和 n 字节 NDEF 消息域，如图 9-6 所示。可接收长度域表示 SNEP 客户端能够接收 SNEP 服务器返回的 NDEF 消息的最大长度。NDEF 消息域中包含一个 NDEF 消息，

用于指示将要被获取的资源。举例来说，该 NDEF 消息可以包含一个 URI Record，用于指示 SNEP 客户端想要从 SNEP 服务器的某条路径或位置上获取 NDEF 数据。若 SNEP 客户端想要获取一个文件，该文件存放在 SNEP 服务器根目录下，那么 SNEP GET 请求消息的信息域中的 NDEF 消息域会包含一个指向该文件路径或位置的 NDEF 消息。

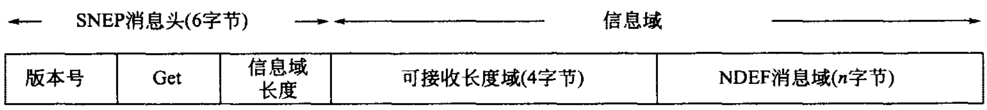


图 9-6　SNEP Get 请求消息

Put 用于 SNEP 客户端请求 SNEP 服务器接收 NDEF 消息，如图 9-7 所示。信息域包含一个 NDEF 消息。

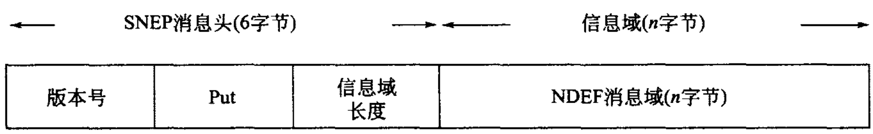


图 9-7　SNEP Put 请求消息

Reject 用于 SNEP 客户端向 SNEP 服务器通告不能获取余下的 SNEP 分段响应消息。SNEP 客户端将不会处理更多的分段消息，Reject 请求消息中不包含信息域。

9.2.3 SNEP 响应消息

SNEP 服务器收到 SNEP 客户端发送的请求消息后，将回复响应消息。SNEP 响应消息如图 9-8 所示。如果信息域长度为 0，响应消息将不包含信息域；或者 SNEP 客户端收到该响应消息后，忽略信息域。

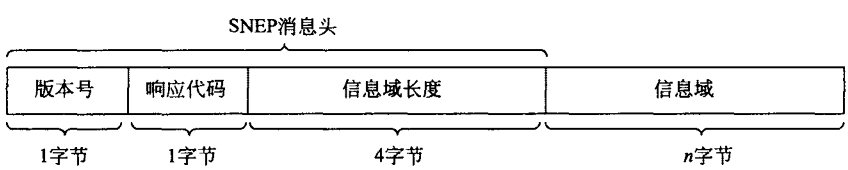


图 9-8　SNEP 响应消息

响应代码用于指示 SNEP 服务器对于请求消息的处理结果，如图 9-9 所示。

与 SNEP 请求消息中的 Continue 类似，SNEP 响应消息中的 Continue 用于 SNEP 服务器请求 SNEP 客户端发送剩余的 SNEP 分段请求消息。该响应消息中不包含信息域。

Success 表示 SNEP 服务器成功地执行了 SNEP 客户端发送的请求消息，并返回 Success 响应消息。如果是对 Get 请求的响应，Success 响应消息中会包含一个 NDEF 消息；如果是对 Put 请求的响应，Success 响应消息中将不包含信息域。

Not Found 表示 SNEP 服务器未能找到匹配的请求。该响应消息中不包含信息域。

Excess Data 表示 SNEP 服务器发现 Get 请求中指定资源的长度超过了最大可接收长度。该响应消息中不包含信息域。

响应代码	名称	描述
00h～7Fh		预留
80h	Continue	继续发送余下的分段
81h	Success	操作成功
82h～BFh		预留
C0h	Not Found	资源未发现
C1h	Excess Data	资源超过了最长的限制
C2h	Bad Request	请求代码错误
C3h～DFh		预留
E0h	Not Implemented	该请求功能未实现
E1h	Unsupported Version	版本号不支持
E2h～FEh		预留
FFh	Reject	不要发送余下的分段

图 9-9　响应代码

Bad Request 表示 SNEP 服务器无法解析请求代码，如语法错误等。该响应消息中不包含信息域。

Not Implemented 表示 SNEP 服务器不支持请求代码表示的功能。该响应消息中不包含信息域。

Unsupported Version 表示 SNEP 服务器不支持或拒绝支持请求消息中指定的 SNEP 协议版本号。该响应消息中不包含信息域。

Reject 用于 SNEP 服务器向 SNEP 客户端通告不能接收余下的 SNEP 分段请求消息。该响应消息中不包含信息域。

9.2.4　SNEP 消息的分段

SNEP 消息可以携带最大 $2^{32}-1$ 个字节的 NDEF 消息。如果 SNEP 请求或响应消息超过 LLC 规定的最大长度，SNEP 消息将分段（Fragment）传输。为了让接收方确定还有多少字节数据要接收，第一个分段必须至少包含 SNEP 消息头。消息头中包含信息域长度，接收方可以此判断是否接收到完整的 SNEP 消息。图 9-10 描述了一个 SNEP 消息被分成三段传输。需要注意的是，即便是 LLC 层能够承载一个 SNEP 消息，SNEP 也可以选择分段传输。

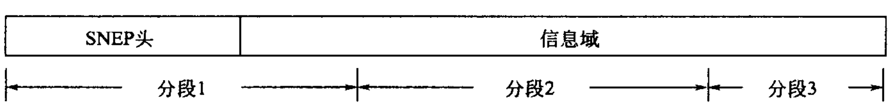


图 9-10　SNEP 消息的分段

接收分段 SNEP 消息的接收方在接收完第一个分段后，向发送方发送 Continue 消息（请求或响应），说明自己能够接收余下的分段消息。如果发送方收到 Continue 消息，将继续发送余下的分段消息，接收方不再确认余下的分段消息。图 9-11 描述了 SNEP 请求消息和 SNEP 响应消息的分段。

图 9-12 描述了 SNEP 客户端向 SNEP 服务器发送的 Get 请求消息，用于获取一个 NDEF 消息。该 NDEF 消息超过了 LLC 层所能承载的最大长度，因此响应消息被分段

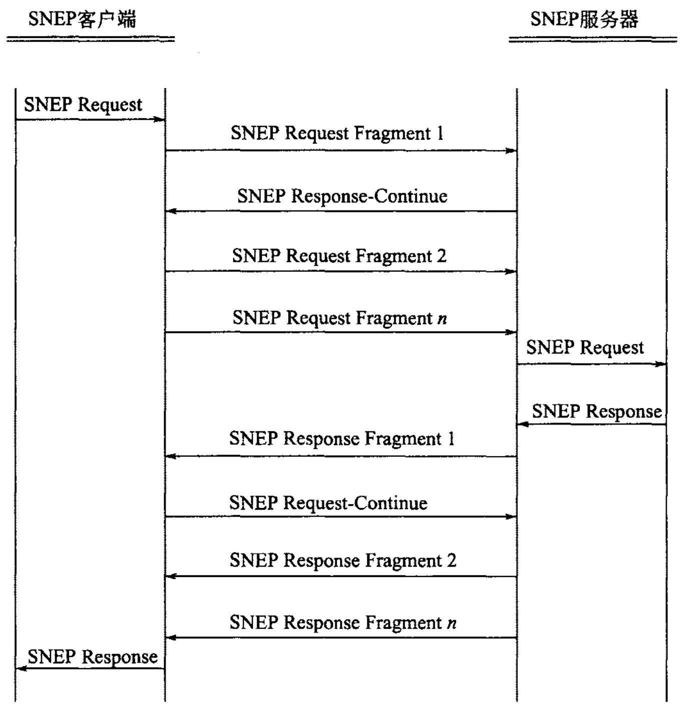


图 9-11 SNEP 请求/响应消息的分段

传输。当 SNEP 客户端收到第一个分段响应消息后，发现 SNEP 头中的信息域长度值大于响应消息的实际值，于是认为这是一个分段响应消息的第一个分段。如果 SNEP 客户端能够接收余下的分段响应消息，SNEP 客户端将发送 Continue 请求消息来响应第一个分段响应消息。SNEP 服务器收到该 Continue 请求消息后，继续发送余下的分段响应消息。

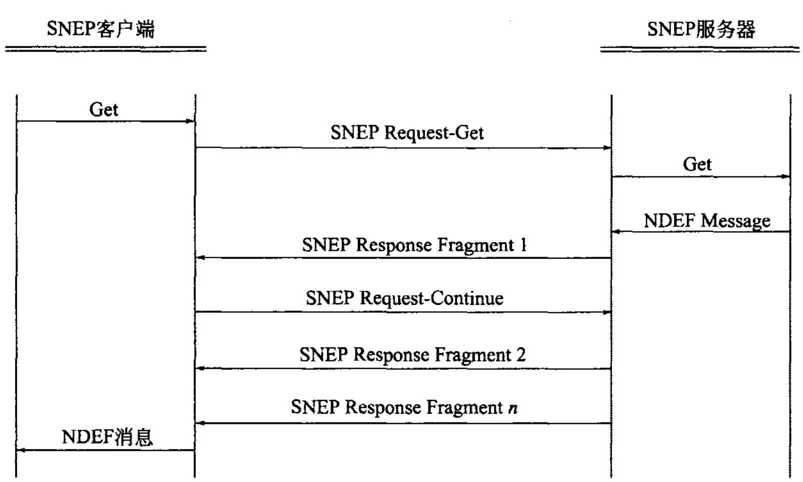


图 9-12 SNEP Get 请求消息的分段

图 9-13 以 SNEP Put 请求消息为例，描述了请求消息分段的示例。

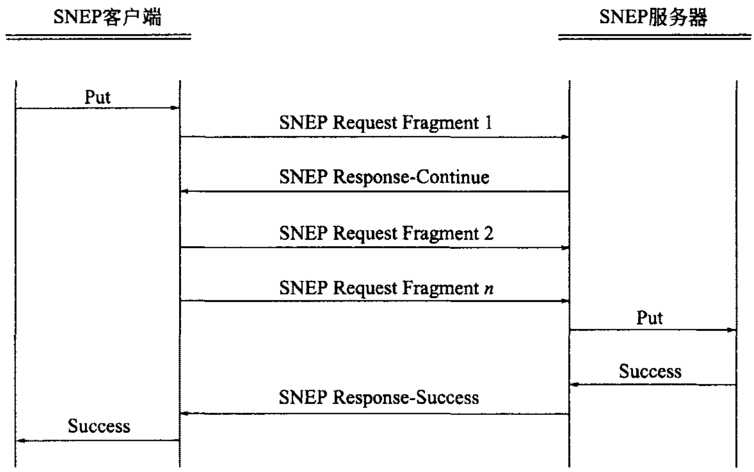


图 9-13　SNEP Put 请求消息的分段

9.3 小结

本章介绍了 SNEP 协议的工作原理。首先简要介绍了 SNEP 规范，SNEP 是一个应用层协议，用于点对点通信的两个 NFC 设备间交互 NDEF 数据，SNEP 协议采用请求/响应架构；然后详细介绍了 SNEP 请求消息/响应消息，如 Put、Get、Continue 等消息的功能和格式；最后举例说明了 SNEP 的分段机制。

第10章 NCI规范

10.1 NCI概述

NCI(NFC Controller Interface) 规范[1]定义了 NFCC(NFC Controller，NFC 控制器)与 DH(Device Host，主机) 之间的接口。NFCC 是指实现 NFC 数据收发，以及运行 NFC 协议栈的实体。DH 负责 NFCC 的管理，如初始化、配置、电源管理等。以具有 NFC 功能的手机为例，NFCC 是指手机中实现 NFC 功能的 NFC 芯片，DH 为手机的处理器。NCI 定义手机 NFC 芯片与处理器之间通信的接口。这里所说的接口不是物理接口，而是 NFCC 与 DH 之间的逻辑接口。DH 和 NFCC 之间通过 SPI、I^2C、UART、USB 等物理层传输技术连接。NCI 规范包含一个传输映射层，将 NCI 规范中的命令、响应及数据适配进不同的物理层传输协议，使 NCI 不依赖物理层传输技术。

NCI 是 NFCC 和 DH 之间通信的逻辑接口规范。NCI 规范的主要目的是标准化 DH 和 NFCC 之间的通信接口。DH 通过标准化接口来控制不同厂商的 NFC 芯片，NCI 屏蔽了不同厂商 NFC 芯片实现的细节以及差异化。在 Android 4.2 之前，Android 系统中提供的

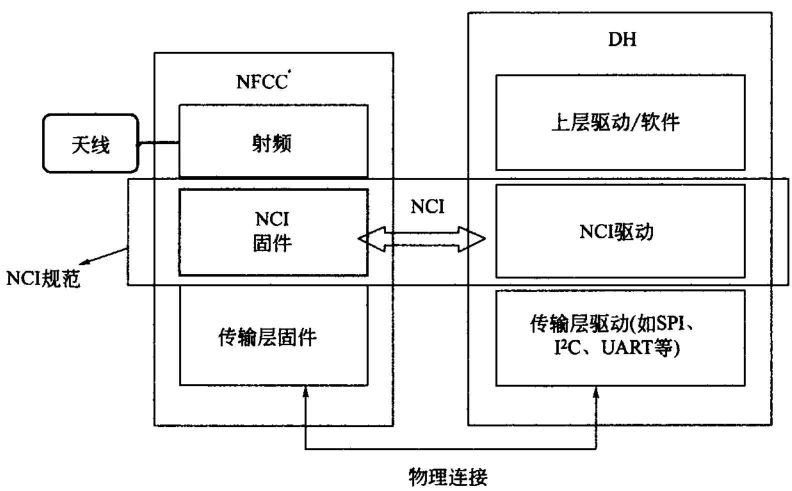


图 10-1 运行在 DH 和 NFCC 上的 NCI

[1] NFC Forum. NFC Controller Interface (NCI) Specification Technical Specification，NCI 1.0.

NFC 协议栈以及芯片参考设计是基于 NXP 的 NFC 芯片。在 4.2 之后，谷歌公司和博通公司合作，采用新的 NFC 协议栈并加入对 NCI 的支持。

NCI 分为两个部分，分别运行在 DH 和 NFCC 上，如图 10-1 所示。

其中，DH 中运行 NCI 驱动，为上层软件提供控制 NFCC 的接口，如将上层软件的请求转换为 NCI 命令向 NFCC 发送。运行在 NFCC 上的 NCI 固件实现了 NFCC 侧的 NCI 协议部分，如解析 DH 发送的 NCI 命令，返回响应消息并执行命令操作。

10.2 NCI 架构

10.2.1 架构

NCI 架构如图 10-2 所示。NCI 架构中包含 3 类组件：NCI 核心、传输映射层以及 NCI 功能模块。

其中，NCI 核心定义了 DH 和 NFCC 之间基本的通信功能，包括 DH 和 NFCC 之间交互的命令、响应和数据消息。

传输映射层在上一节中介绍过，其主要作用是根据 DH 和 NFCC 之间的物理连接方式，将 NCI 控制消息和数据消息适配进相应的物理传输协议中。

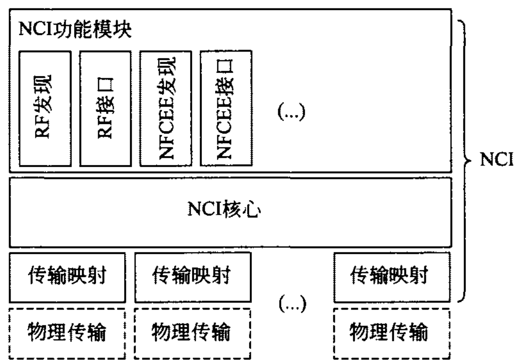


图 10-2 NCI 架构

NCI 功能模块位于 NCI 核心之上，完成特定的功能，如允许 DH 配置 NFCC 来发现远端 NFC 端点（Reomte NFC Endpoints，本章中为了与 NCI 协议描述一致，将周围 NFC 设备和标签统称为远端 NFC 端点），并与远端 NFC 端点通信；同时，NFC 功能模块为 DH 提供了配置 NFCEE 和与 NFCEE 通信的功能。

NFCEE(NFC Execution Environment，NFC 执行环境）是一个逻辑实体，为 NFC 应用程序提供执行环境。NFCEE 的具体实现形式可以是 SD 卡、SIM 卡等。以 SIM 卡为例，SIM 卡中运行着操作系统，如 Javacard OS，为 NFC 应用程序提供运行和执行的环境。在金融应用，特别是在 NFC 移动支付中，银行的支付应用一般被安装运行在 SIM 卡中或 SD 卡中，此时 NFCEE 也称为 SE（Security Element，安全单元）。关于移动支付和安全单元的细节将在第 14 章介绍。另外，DH 可以运行 NFC 应用程序，因此也可以作为一个 NFCEE，称为 DH-NFCEE。在 NCI 规范中，根据 NFCC 与 NFCEE 之间的连接关系，定义了两种架构，即单主架构（Single Host）和多主架构（Multiple Host），如图 10-3 所示。这里的单主和多主中的 Host 是针对 NFCC 来说的。在单主架构中，DH 作为 NFCC 的唯一主机，对 NFCC 进行初始化和管理，并通过 NFCC 与远端 NFC 端点通信。在多主架构中，NFCC 连接 DH 和 NFCEE 两个主机，由于 NFCEE 的实体形式可以是 SD 卡或 SIM 卡等，本身 SD 卡或 SIM 卡就具有处理器以及相应的操作系统，因此也作为一个主机来控制和管理 NFCC。这时，NFCC 作为共享资源供 DH 和 NFCEE 使用。在多主架构中，NFCC 可以连接多个 NFCEE。

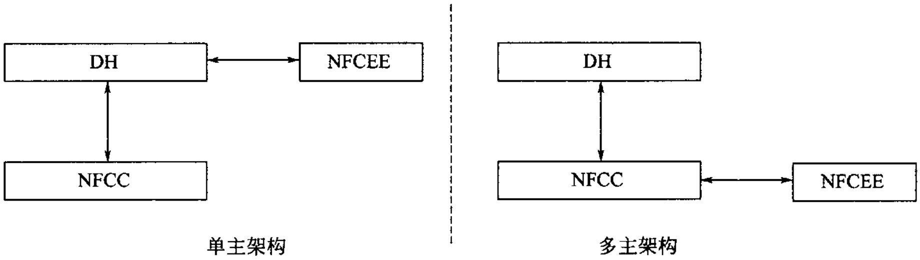


图 10-3　单主架构与多主架构

需要注意的是，在大部分情况下，多主架构中的 DH 和 NFCEE 之间有连接关系。但是不同标准化组织的着眼点不同，NFC Forum 中以 NFC 芯片厂商为主导方，因此在 NCI 规范中以 NFCC 为核心，DH 和 NFCEE 都与 NFCC 相连，DH 访问 NFCEE 也要通过 NFCC；而在其他标准化组织的规范中，一般是 DH 直接控制和管理 NFCEE。

NCI 功能模块中的某些模块是必要的，有些是可选的。在 NFC 功能模块之间也存在一定的依赖关系。举例来说，与远端 NFC 端点通信的 NFC 功能模块依赖于 RF 发现功能模块，因为只有通过 RF 发现模块才能发现远端 NFC 端点。NFCEE 和 NFCC 之间的通信方式及连接方式不在 NCI 规范范围内，但是 DH 可以通过 NFCC 管理 NFCEE 并与 NFCEE 通信，因此 NCI 中定义了 NFCEE 发现和 NFCEE 接口两个功能模块。

10.2.2 通信流程

NCI 基本通信流程如图 10-4 所示。

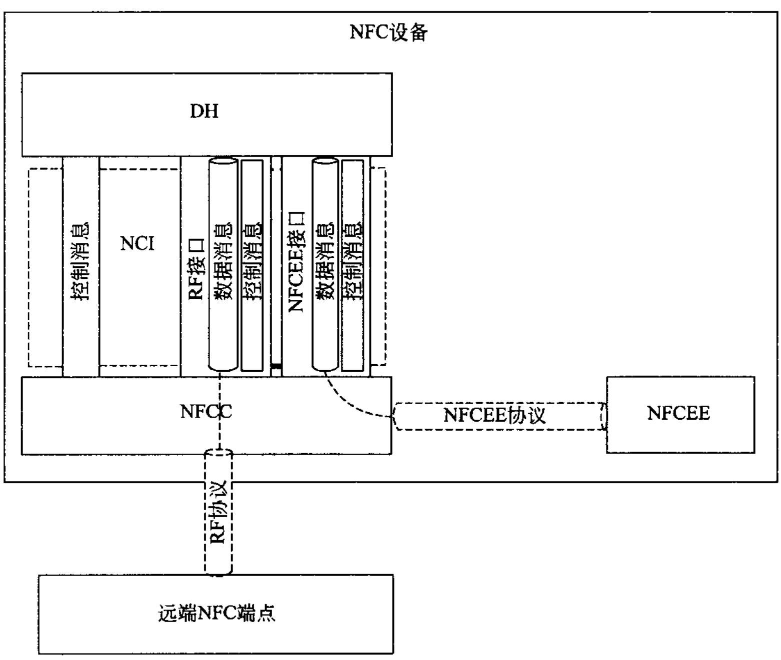


图 10-4　NCI 基本通信流程

DH 通过控制消息来控制和配置 NFCC。控制消息包括命令、响应以及通告。命令是由 DH 发送给 NFCC 的，响应和通告则由 NFCC 发送给 DH。NCI 核心定义了一个基本的控制

消息集合，如设置和获取NFCC配置参数等。其他涉及功能模块的控制消息由各个NFC功能模块定义。

数据消息在DH和远端NFC端点，即周围的NFC目标设备/标签之间交互，或者在DH和NFCEE之间交互。DH通过NFCC向远端NFC端点或NFCEE发送数据消息，并且DH通过NFCC接收远端NFC端点或NFCEE发送的数据。在DH通过NFCC收发数据之前，必须和NFCC建立逻辑连接。在NFCC初始化时，会默认建立好一个逻辑连接，称之为静态射频连接。该连接用于RF通信。其他动态连接需要在与远端NFC端点或NFCEE通信之前创建。逻辑连接为DH与NFCC之间的数据消息提供流控制功能。一般来说，NFCC的处理能力以及存储能力弱于DH，因此流控只用于从DH到NFCC的数据消息。

NCI中定义了两类接口：RF接口和NFCEE接口。RF接口用于DH和远端NFC端点通信，NFCEE接口用于DH与NFCEE通信。每一个接口都支持特定的协议。DH与远端NFC端点通信的协议称为RF协议，DH与NFCEE通信的协议称为NFCEE协议。举例来说，RF协议可以是第4章提到的NFC-DEP协议、ISO-DEP协议、类型1/2/3标签通信协议等，也可以是第5章提到的LLCP协议。可以看出，RF协议的选择取决于两个通信的NFC设备或设备与标签之间所支持的技术与相应的Digital协议。由于RF协议不同，因此对应不同的RF接口。RF接口根据RF协议分为NFC-DEP RF接口、ISO-DEP RF接口、Frame RF接口等。这里需要进一步说明RF接口与RF协议之间的关系。RF协议实际上是NFCC与远端NFCC之间采用的通信协议，但是对于DH来说，该RF协议没有运行在DH上，DH无法“看到”它，因此NCI定义了RF接口的概念。DH通过RF接口控制对应的RF协议。从RF接口的概念来看，“RF接口是用于DH和远端NFC端点通信的接口”，这里强调的是，DH只有通过RF接口才能和远端NFC端点通信。

RF接口和RF协议有映射关系。映射关系是DH初始化NFCC时通过配置命令配置给NFCC的。DH向NFCC发送RF发现命令，让NFCC执行RF发现过程；通过RF发现过程，NFCC把收集到的远端NFC端点的信息反馈给DH，如远端NFC端点支持的RF协议等信息。如果在RF发现过程中只发现了一个远端NFC端点，且该远端NFC端点只支持一种RF协议，NFCC将根据RF接口与RF协议的映射表自动激活对应的RF接口。激活的过程实际上就是NFCC向DH发送一个RF接口激活通告消息。如果在RF发现过程中发现了多个远端NFC端点，或者是一个远端NFC端点支持多个RF协议，NFCC将向DH通告所有端点的信息。如果一个远端NFC端点支持多个RF协议，NFCC将根据其支持的RF协议，将其看成多个远端NFC端点。举例来说，如果一个远端NFC端点既支持NFC-DEP协议，也支持ISO-DEP协议，NFCC将其看成两个不同的远端NFC端点，分别支持NFC-DEP协议和ISO-DEP协议。当DH收到NFCC发送的远端NFC端点列表后，可以选择其中一个通信。选择的过程将激活对应的RF接口。需要注意的是，NCI并没有定义选择的策略，即选择哪一个端点的规则，只是提供了选择的方法。因为如何选择，是由上层软件或用户决定的，不在NCI规范的考虑范围内，10.4节将介绍Android中如何选择远端NFC端点。

RF接口激活后，DH就可以通过RF接口向NFCC发送控制消息，以及与远端NFC端点进行数据交互。这里的控制消息在前面提到过，是NCI功能模块中的RF接口模块所属的控制消息，而不是NCI核心发送的基本控制消息。通信结束后，需要对RF接口去激活。去激活可以由DH或NFCC发起。关于RF接口的具体内容，将在10.5节介绍。NFCEE接口与RF接口原理类似，NFCC与NFCEE之间的通信协议对应NFCEE接口，DH通过NFCEE接口与NFCEE进行数据交互。

同样，DH 可以通过 NFCEE 发现模块来获取 NFCC 上连接的 NFCEE 的相关信息。在 NFCEE 发现过程中，NFCC 为每一个 NFCEE 分配一个标识符。当 DH 想与某个 NFCEE 通信时，使用相应的标识符和指定 NFCEE 协议创建逻辑连接。此时，将自动激活相应的 NFCEE 接口。这一点与 RF 接口不同。由于 NFCEE 接口与 NFCEE 协议是一一对应的关系，因此无需像 RF 接口那样配置接口和协议的映射表。NFCEE 接口激活后，DH 就可以通过 NFCEE 接口与 NFCEE 通信了。如图 10-4 所示，DH 可以通过 NFCEE 接口向 NFCC 发送控制消息，与 NFCEE 进行数据交互。这里所指的控制消息，是指 NCI 功能模块的 NF-CEE 接口模块所属的控制消息。关闭与 NFCEE 之间的逻辑连接，将去激活 NFCEE 接口。

在 RF 发现过程和 NFCEE 发现过程中，NFCC 为远端 NFC 端点和 NFCEE 动态地分配标识符。远端 NFC 端点的标识符称为 RF 发现 ID（RF Discovery ID），NFCEE 的标识符称为 NFCEE ID。前面提到的 DH-NFCEE 的 ID 值预留为 0。

下面将分别介绍 NCI 功能模块。

10.3 NCI 核心框架

10.3.1 核心框架概述

NCI 核心主要包括以下功能：

① 定义了 DH 和 NFCC 之间基本的命令、响应和通告消息。

② 命令/响应消息的流控机制。

③ 数据消息的逻辑连接

④ 从 DH 到 NFCC 的数据消息流控机制。

⑤ 控制消息和数据消息的分段重组机制。

⑥ NFCEE 寻址。

⑦ 远端 NFC 端点寻址。NCI 使用逻辑连接支持 DH 与 NFCEE、远端 NFC 端点之间的通信。

⑧ 对 NFCC 的重置、初始化和配置。

⑨ 控制消息的异常处理。

从其主要功能来看，NCI 核心实际上就是为 DH 和 NFCC 之间提供基本的通信功能，同时为 DH 与 NFCEE、DH 与远端 NFC 端点之间提供基本的通信支持。

10.3.2 NCI 消息格式

NCI 定义了控制消息和数据消息的格式，如图 10-5 所示。

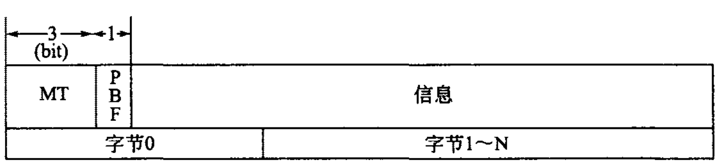


图 10-5 NCI 消息格式

其中，MT(Message Type) 表示消息类型，取值如表 10-1 所示。

表 10-1 消息类型

MT	描 述	MT	描 述
000b	数据消息	011b	控制消息：通告
001b	控制消息：命令	100b～111b	预留
010b	控制消息：响应		

PBF(Packet Boundary Flag) 用于分段和重组功能。如果 PBF 被设置为 0，表示该消息为一个完整的消息，或者该消息为分段消息中的最后一个分段。如果设置为 1，表示为分段消息，并且不是最后一个分段消息。

信息域中包含控制或数据消息的内容。

10.3.3 NCI 控制消息

10.3.3.1 控制消息格式

NCI 控制消息包括命令、响应和通告。控制消息的交互如图 10-6 所示。DH 向 NFCC 发送命令，NFCC 接收到命令后向 DH 回复响应。NFCC 也可以向 DH 发送通告消息，如向 DH 报告 RF 发现过程中获取的远端 NFC 端点信息。

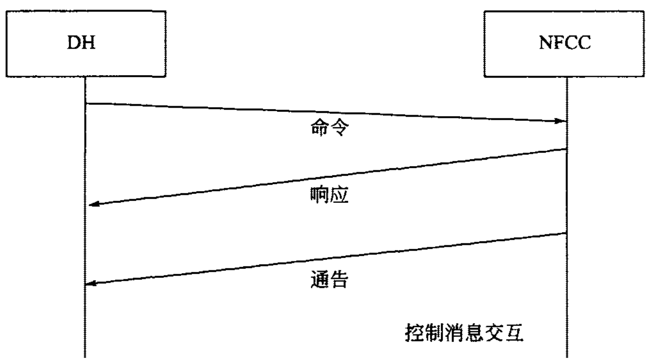


图 10-6 NCI 控制消息的交互

图 10-7 描述了控制消息的格式。

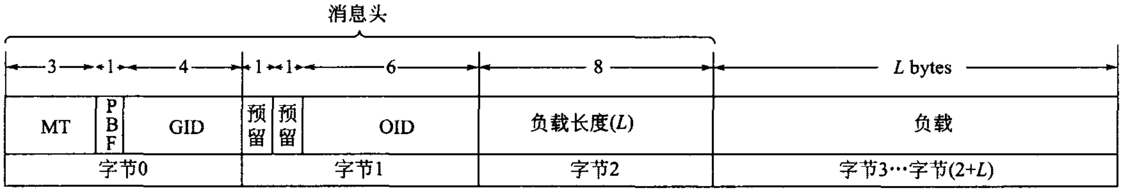


图 10-7 控制消息格式

在 NCI 规范中，根据控制消息的功能，将其归类到不同的功能组。功能组用 GID (Group ID) 表示，如表示 NCI 核心的控制消息组、RF 管理功能的控制消息组、NFCEE 管理的控制消息组等。

在 GID 所指示的功能组中，OID(Opcode ID) 唯一地表示一个命令、响应、通告的集

合。这里所说的集合，是指包含命令和响应，或者是包含一个通告，或者是包含命令、响应和通告。举例来说，在 GID 表示的 NCI 核心功能中，OID＝0000b 表示重置 NFCC 的命令、响应和通告，OID＝0001b 表示初始化 NFCC 的命令和响应，OID＝1000b 表示接口错误的通告。对于 GID 和 OID 的取值，参考表 10-54。

负载长度 L 表示负载域所能承载的字节数。控制消息最大所能承载的负载长度为 255 字节。一般情况下，DH 可以接收负载为 255 字节的控制消息；但是对于 NFCC 来说，由于处理能力及存储限制，无法接收负载 255 字节的控制消息，因此 DH 需要对控制消息分段。NFCC 接收到分段的控制消息后进行重组。

10.3.3.2 控制消息的流控

对于控制消息来说，NCI 没有定义特殊的流控机制，只是要求 DH 发送一条命令后，只有在收到该命令的响应后才能发送下一条命令。尽管 DH 和 NFCC 之间的物理链路一般是可靠的，但 DH 还是维护了一个计时器。如果计时器超时还未收到命令的响应，DH 将重发该命令或发送下一条命令。对于 NFCC 来说，NFCC 回复响应后，将等待接收下一条命令。DH 应当能够在任何时候接收 NFCC 发送的通告。

10.3.3.3 重置 NFCC

在 NFCC 上电启动后，DH 向 NFCC 发送 CORE _ RESET _ CMD 命令重置 NFCC。重置命令 CORE _ RESET _ CMD 的格式如表 10-2 所示。

表 10-2 CORE _ RESET _ CMD

负载域	长度	值/描述	
重置类型	1 字节	0x00	保持现有配置 重置 NFCC 并保持现有 RF 配置
		0x01	重置配置 重置 NFCC，包括重置 RF 配置
		0x02～0xFF	预留

CORE _ RESET _ CMD 命令的负载域中包含 1 字节的重置类型。重置类型取值不同，对 NFCC 的重置方式也不尽相同。重置类型取值为 0x00 时，表示 DH 希望保持现有 RF 配置，但是对其他配置参数重置。这里的 RF 配置是指进行 RF 发现及 RF 通信时需要的配置参数，包括侦听模式路由表、RF 接口映射表，如表 10-55 中所示的参数。关于侦听模式路由表，将在 10.8 节介绍。当重置类型为 0x01 时，表示 DH 将重置 NFCC 的所有配置参数。

NFCC 收到 CORE _ RESET _ CMD 后，向 DH 发送响应消息 CORE _ RESET _ RSP，其格式如表 10-3 所示。所有的响应消息中都包含状态参数，如表 10-56 所示。状态参数表示 NFCC 收到命令后执行的状态，如 STATUS _ OK 表示正在执行命令，STATUS _ REJECT 表示 NFCC 无法执行该命令等。NCI 版本表示 NFCC 支持的版本号。目前 NCI 规范的版本号为 1.0，因此该参数取值为 0x01。配置状态表示 NFCC 执行重置命令的结果。当配置状态为 0x01 时，表示 NFCC 移除了侦听模式路由表中的所有内容，删除了 RF 接口映射表，并将参考表 10-55 中的所有参数重置为默认值。需要注意的是，当 CORE _ RESET _ CMD 中的重置类型设置为 0x00 时，NFCC 返回的 CORE _ RESET _ RSP 中的配置状态可以是 0x00，也可以是 0x01，主要依赖于 NFCC 的实现。

表 10-3 CORE _ RESET _ RSP

负 载 域	长 度	值/描述
状态	1 字节	见表 10-56
NCI 版本	1 字节	目前 NCI 版本号取值为 0x01
配置状态	1 字节	0x00:NCI RF 配置已被保持 0x01:NCI RF 配置已被重置 0x02～0xFF:预留

当 NFCC 出现内部错误时，NFCC 也可以重置自己，即不通过 CORE _ RESET _ CMD 触发重置。在这种情况下，NFCC 重置后将向 DH 发送 CORE _ RESET _ NTF 通告消息，如表 10-4 所示。原因代码表示重置的原因。目前 NCI 规范中没有列举具体的原因。配置状态的含义与 CORE _ RESET _ RSP 中一致。

表 10-4 CORE _ RESET _ NTF

负 载 域	长 度	值/描述	
原因代码	1 字节	0x00	未详细说明
		0x01～0x9F	预留
		0xA0～0xFF	私有
配置状态	1 字节	0x00:NCI RF 配置已被保持 0x01:NCI RF 配置已被重置 0x02～0xFF:预留	

10.3.3.4 初始化 NFCC

NFCC 被重置后，DH 紧接着发送初始化命令，对 NFCC 进行初始化操作。初始化命令 CORE _ INIT _ CMD 的格式如表 10-5 所示，CORE _ INIT _ CMD 中没有包含任何负载内容。

表 10-5 CORE _ INIT _ CMD

负 载 域	长 度	值/描述
负载域为空		

NFCC 收到 CORE _ INIT _ CMD 后，向 DH 返回 CORE _ INIT _ RSP，表示 NFCC 开始执行初始化操作，如表 10-6 所示。如果初始化操作成功执行，状态参数将设置为 STATUS _ OK；如果不能执行初始化命令，状态参数设置为 STATUS _ FAILED。NFCC 特性参数包括 4 字节，表示 NFCC 支持的一些特性说明。其中，字节 1 和字节 2 表示侦听模式路由表对路由方式的支持情况，具体内容在后续章节详细介绍。

10.3.3.5 配置 NFCC

NFCC 初始化后，所有的参数都被设置为默认值，但是 DH 还可以通过配置命令 CORE _ SET _ CONFIG _ CMD 修改这些参数，如表 10-7 所示。参数由 ID 值标识。参数 ID 值如表 10-55 所示。

表 10-6 CORE _ INIT _ RSP

负载域	长度	值/描述
状态	1字节	见表10-56
NFCC特性	4字节	字节0 b2～b1表示发现配置模式。如果设置为00，表示只能由DH配置NFCC；如果设置为01，NFCC可以由DH或NFCEE配置，NFCC可以管理或合并多套RF配置参数 b0表示RF发现命令RF_DISCOVER_CMD是否支持RF发现频率的设置。如果b0＝1，表示支持；如果b0＝0，表示不支持 b3～b7，任意取值 字节1 b3＝1，表示侦听模式路由表支持基于AID的路由方式 b2＝1，表示侦听模式路由表支持基于协议的路由方式 b1＝1，表示侦听模式路由表支持基于技术的路由方式 字节2 b1＝1，表示侦听模式路由表支持关机状态(Switched Off)。 b0＝1，表示侦听模式路由表支持无电状态(Battery Off)。 字节3 预留，取值全0
支持RF接口的个数	1字节	表示NFCC支持的RF接口个数
支持的RF接口[1..n]	1字节	RF接口类型，见表10-19
最大的逻辑连接数	1字节	0x00～0x0E：NFCC支持最大的动态逻辑连接数 0x0F～0xFF：预留
最大的路由表容量	2字节	侦听模式路由表能够支持的最大数据量。如果NFCC不支持侦听模式路由表，返回0x0000
最大控制消息负载长度	1字节	NFCC能接收的最大控制消息负载长度，取值为32～255
巨大参数的最大字节数	2字节	PB_H_INFO和LB_H_INFO_RESP合在一起的最大字节数。PB_H_INFO表示ATTRIB命令中的信息域，LB_H_INFO_RESP表示ATTRIB响应中的信息域
厂商ID	1字节	如果NFCC不支持该参数，则返回0x00
厂商信息	4字节	如果NFCC不支持该参数，则返回4字节的0x00

表 10-7 CORE _ SET _ CONFIG _ CMD

负载域	长度	值/描述		
参数的个数	1字节	配置命令中携带的参数个数		
参数[1…n]	m+2字节	ID	1字节	参数的ID值，见表10-55
		长度	1字节	参数的长度。如果长度为0x00，表示DH将该参数设置为默认值
		值	m字节	配置参数的取值

NFCC收到配置命令后将回复CORE _ SET _ CONFIG _ RSP，如表10-8所示。其中的状态参数指示配置是否成功，STATUS _ OK表示所有的配置参数被设置成功。如果NFCC不支持相应参数，状态参数为STATUS _ INVALID _ PARAM，并且在CORE _ SET _

CONFIG _ RSP 中携带这些无效参数的 ID 值。

表 10-8 CORE _ SET _ CONFIG _ RSP

负 载 域	长 度	值/描述
状态	1 字节	见表 10-56
参数的个数	1 字节	如果状态参数为 STATUS_INVALID_PARAM，那么参数的个数为无效参数的个数
参数 ID[0…n]	1 字节	无效参数的 ID 值

DH 向 NFCC 发送 CORE _ GET _ CONFIG _ CMD 来获取 NFCC 的参数列表，如表 10-9 所示。

表 10-9 CORE _ GET _ CONFIG _ CMD

负 载 域	长 度	值/描述
参数的个数	1 字节	参数的个数
参数 ID[1…n]	1 字节	参数的 ID 值

NFCC 收到 CORE _ GET _ CONFIG _ CMD 后，向 DH 回复 CORE _ GET _ CONFIG _ RSP，如表 10-10 所示。如果 NFCC 能够响应所有的参数，那么状态参数设置为 STATUS _ OK，并包括所有参数值。如果 NFCC 不能支持某些参数，那么状态参数设置为 STATUS _ INVALID _ PARAM，只包括无效的参数列表。

表 10-10 CORE _ GET _ CONFIG _ RSP

<table>
<tr><th>负 载 域</th><th>长 度</th><th colspan="3">值/描述</th></tr>
<tr><td>状态</td><td>1 字节</td><td colspan="3">见表 10-56</td></tr>
<tr><td>参数的个数</td><td>1 字节</td><td colspan="3">配置命令中携带的参数的个数</td></tr>
<tr><td rowspan="3">参数[1…n]</td><td rowspan="3">m+2 字节</td><td>ID</td><td>1 字节</td><td>参数的 ID 值，见表 10-55</td></tr>
<tr><td>长度</td><td>1 字节</td><td>参数的长度</td></tr>
<tr><td>值</td><td>m 字节</td><td>参数的取值</td></tr>
</table>

10.3.3.6 逻辑连接管理

10.2.2 节介绍过，RF 接口或 NFCEE 接口用于 DH 与远端 NFC 端点或 NFCEE 通信。接口激活后，DH 通过接口向 NFCC 发送控制消息，以及与远端 NFC 端点或 NFCEE 进行数据交互，如图 10-4 所示。实际上，通过接口进行数据交互前还有一个必要的步骤，那就是创建逻辑连接。逻辑连接分为两类，即静态 RF 连接（Static RF Connection）和动态逻辑连接（Dynamic Logical Connection）。

静态 RF 连接只是针对 RF 接口来说的，在 NFCC 初始化完成后就存在，是不需要创建的，但必须在 RF 接口激活后才能使用。动态逻辑连接用于 RF 接口和 NFCEE 接口，需要 DH 向 NFCC 发送 CORE _ CONN _ CREATE _ CMD 来动态创建。对于 RF 接口来说，在 NCI 1.0 规范中只支持静态 RF 连接的使用，不支持动态逻辑连接。对于 NFCEE 接口来说，DH 与 NFCEE 进行数据交互前，必须建立动态逻辑连接。

逻辑连接实际上为 DH 和 NFCC 的数据交互提供了一些传输参数，如数据交互过程中的最大数据消息长度、流控参数等。每一个逻辑连接都通过连接 ID 来标识，是由 NFCC 分配的；静态 RF 连接的 ID 值无需分配，默认为 0。

DH 通过发送 CORE _ CONN _ CREATE _ CMD 来创建逻辑连接，如表 10-11 所示。该命令中指定了目标类型，以及与该目标类型相关的参数。目标类型用于表示逻辑连接上的数据的目的地。目标类型取值为 0x01 时，称为 NFCC 回环连接。该逻辑连接上的数据只在 DH 和 NFCC 之间传输，不会向远端 NFC 端点或 NFCEE 发送。该逻辑连接的主要目的是用于对 NFCC 的测试。当目标类型取值为 0x02 时，该逻辑连接上的数据在 DH 和远端 NFC 端点之间传输。当目标类型取值为 0x03 时，该逻辑连接上的数据在 DH 和 NFCEE 之间传输。

表 10-11　CORE _ CONN _ CREATE _ CMD

<table>
<tr><th>负　载　域</th><th>长　度</th><th colspan="3">值/描述</th></tr>
<tr><td>目标类型</td><td>1 字节</td><td colspan="3">0x00：预留
0x01：NFCC Loop Back，用于 DH 创建 NFCC 回环连接，主要目的用于测试
0x02：远端 NFC 端点
0x03：NFCEE
0x04～0xC0：预留
0xC1～0xFE：私有
0xFF：预留</td></tr>
<tr><td>目标相关的参数个数</td><td>1 字节</td><td colspan="3">根据目标类型，与目标类型相关的参数个数</td></tr>
<tr><td rowspan="3">目标相关的参数[0…n]</td><td rowspan="3">m+2 字节</td><td>类型</td><td>1 字节</td><td>0x00：值为 2 字节。第一个字节表示 RF 发现 ID，第二个字节表示 RF 协议
0x01：值为 2 字节。第一个字节表示 NFCEE ID，第二个字节表示 NFCEE 协议
0x02～0x9F：预留
0xA0～0xFF：私有</td></tr>
<tr><td>长度</td><td>1 字节</td><td>值的长度</td></tr>
<tr><td>值</td><td>m 字节</td><td>当类型为 0x00 时，值为 2 字节。第一个字节表示 RF 发现 ID，第二个字节表示 RF 协议
当类型为 0x01 时，值为 2 字节。第一个字节表示 NFCEE ID，第二个字节表示 NFCEE 协议</td></tr>
</table>

根据不同的目标类型，该命令携带不同的参数。当逻辑连接的类型为远端 NFC 端点时，参数为 2 字节，第一个字节为 RF 发现 ID，第二个字节表示 RF 协议。NFCC 在执行 RF 发现过程中，为每一个远端 NFC 端点分配一个 ID 值。第二个字节表示与远端 NFC 端点之间采用 RF 协议。当逻辑连接的类型为 NFCEE 时，参数也为 2 字节，第一个字节为 NFCEE ID，第二个字节为 NFCEE 协议。当 NFCC 执行 NFCEE 发现过程中，为每一个连接在 NFCC 上的 NFCEE 分配一个 ID 值。前面提到过，没有专门的激活命令来激活 NFCEE 接口，当 DH 向 NFCC 建立目标类型为 NFCEE 的逻辑连接时，NFCEE 接口被激活。

当 NFCC 收到 CORE _ CONN _ CREATE _ CMD 后，根据目标参数判断是否接收该连接请求，并返回 CORE _ CONN _ CREATE _ RSP，其中的状态参数表明是否接受 DH 的连接请求。如果状态参数为 STATUS _ OK，表示 NFCC 接受 DH 的连接请求；如果状态参数为 STATUS _ REJECTED，表示 NFCC 拒绝 DH 的连接请求。CORE _ CONN _ CREATE _ RSP 如表 10-12 所示。最大数据消息负载长度表示该逻辑连接上数据消息负载的最大长度限制。

Retro Workshop 果粉工作室

表 10-12 CORE _ CONN _ CREATE _ RSP

负载域	长度	值/描述
状态	1 字节	见表 10-56
最大数据消息负载长度	1 字节	取值范围 1～255
初始的 Credits 值	1 字节	取值范围 0x00～0xFE
连接 ID	1 字节	高 4 位：预留低 4 位：0000b，用于静态 RF 连接；0001b～1111b，由 NFCC 动态分配

连接 ID 是 NFCC 为该逻辑连接分配的 ID 值，用于唯一地标识该连接。在后面介绍的数据消息格式中，可以看到数据消息的头中包含所属连接的 ID 值，表示该数据消息属于哪个逻辑连接。当 NFCC 收到携带连接 ID 的数据消息后，根据连接 ID 值知道逻辑连接的目标类型及目标参数（如 RF 发现 ID/RF 协议，NFCEE ID/NFCEE 协议），于是向该目标发送数据消息。

初始 Credits 值用于数据消息的流控。在 10.2.2 节中提到过，在一般情况下，NFCC 的处理能力以及存储能力弱于 DH，因此流控机制用于从 DH 发送到 NFCC 的数据消息。在 NCI 规范中，从 DH 到 NFCC 的数据消息采用基于 Creidt 的流程机制。数据消息的流控功能是可选的，如果需要使用流控功能，开启该功能。

对于每一个逻辑连接，DH 通过 CORE _ CONN _ CREATE _ RSP 消息从 NFCC 获取初始的 Credit，取值范围为 0x00～0xFE。DH 将 Credit 初始值存放在 NFCC _ Credits _ Avail 参数中，用于判断允许发送到 NFCC 的数据消息的个数。在流控功能开启的情况下，若 DH 想从逻辑连接上发送数据消息，DH 需要检查 NFCC _ Credits _ Avail 是否大于 0。如果大于 0，DH 将发送数据消息，并对 NFCC _ Credits _ Avail 进行减 1 操作。当 NFCC _ Credits _ Avail 为 0 时，DH 不被允许在该逻辑连接上发送任何数据消息。

如果 NFCC 有可供使用的缓存，NFCC 将向 DH 发送通告消息 CORE _ CONN _ CREDITS _ NTF，如表 10-13 所示。DH 收到该通告消息后，更新 NFCC _ Credits _ Avail 参数。

表 10-13 CORE _ CONN _ CREDITS _ NTF

负载域	长度	值/描述		
Entry 的个数	1 字节	表示 Entry 的个数。一个 Entry 由连接 ID 和 Credit 组成		
Entry[1…n]	2 Octet	连接 ID	1 字节	连接 ID
		Credit	1 字节	连接 ID 所对应的 Credit。NFCC 向 DH 发送该 Credit 值，用于 DH 更新 NFCC_Credits_Avail 参数

需要注意的是，静态 RF 连接是不需要通过创建命令建立的。那么，DH 如何获取该逻辑连接的相关参数，如初始的 Credit，最大数据消息负载长度呢？实际上，这些参数同样包含在 RF 接口激活通告消息（RF _ INTF _ ACTIVATED _ NTF）中。可以看出，RF 接口激活时，NFCC 向 DH 发送 RF _ INTF _ ACTIVATED _ NTF 消息，该通告消息中包含与 CORE _ CONN _ CREATE _ RSP 相同的参数。静态 RF 连接使用这些参数值，但并不影响 NFCC 为该 RF 接口上的动态逻辑连接重新定义参数值，这也为逻辑连接的 QoS(Quality of Service) 提供了可能。

数据通信结束后，需要关闭逻辑连接。DH 向 NFCC 发送 CORE _ CONN _ CLOSE _ CMD 命令来关闭指定的逻辑连接，如表 10-14 所示。

表 10-14 CORE _ CONN _ CLOSE _ CMD

负载域	长度	值/描述
连接 ID	1 字节	要关闭逻辑连接的 ID 值

NFCC 收到该 CORE _ CONN _ CLOSE _ CMD 后，向 DH 发送 CORE _ CONN _ CLOSE _ RSP，如表 10-15 所示。如果 NFCC 找到匹配的连接 ID，CORE _ CONN _ CLOSE _ RSP 的状态参数为 STATUS _ OK。如果 NFCC 未能找到匹配的连接 ID，状态参数为 STATUS _ REJECTED。DH 发送完 CORE _ CONN _ CLOSE _ CMD 消息后，将清除所有与该连接相关的资源。NFCC 收到该命令后，也会清除所有与该连接相关的资源。清除资源是指删除该连接的数据缓存，释放连接 ID。对于 DH 来说，还会将 NFCC _ Credits _ Avail 设置为初始值。

表 10-15 CORE _ CONN _ CLOSE _ RSP

负载域	长度	值/描述
状态	1 字节	见表 10-56

10.3.3.7 错误处理

在某些出错的情况下，NFCC 无法回复响应消息，因此 NCI 规范定义了通用错误通告消息和接口错误通告消息，如表 10-16 和表 10-17 所示。通用错误通告消息 CORE _ GENERIC _ ERROR _ NTF 用于通告一些通用的错误情况，但不通告与 NFCEE 或 RF 接口通信相关的逻辑连接错误信息。

表 10-16 CORE _ GENERIC _ ERROR _ NTF

负载域	长度	值/描述
状态	1 字节	见表 10-56

NFCC 向 DH 发送 CORE _ INTERFACE _ ERROR _ NTF 通告消息来通告与逻辑连接相关的错误。

表 10-17 CORE _ INTERFACE _ ERROR _ NTF

负载域	长度	值/描述
状态	1 字节	见表 10-56
连接 ID	1 字节	错误发生的逻辑连接

10.3.4 NCI 数据消息

DH 和 NFCEE 或 DH 和远端 NFC 端点的数据交互需要通过 NFCC 中转，如图 10-8 所示。

在 DH 与 NFCC 之间的数据称为 NCI 数据消息，由 NCI 规范制定。NFCC 与 NFCEE 之间的数据交换遵循 NFCEE 协议，NFCC 与远端 NFC 端点的数据交换遵循所采用的 RF 协议，如 NFC-DEP、ISO-DEP 等。DH 和 NFCEE 或 DH 与远端 NFC 端点在数据交互之前，DH 与 NFCC 之间需要建立逻辑连接。NCI 数据消息通过逻辑连接在 DH 和 NFCC 之间交换，换句话说，NCI 数据消息的头中包含连接 ID，用于指示所属的逻辑连接。DH 通过逻辑连接向 NFCC 发送数据，数据被封装成 NCI 的数据消息格式，当 NFCC 收到后，通过数

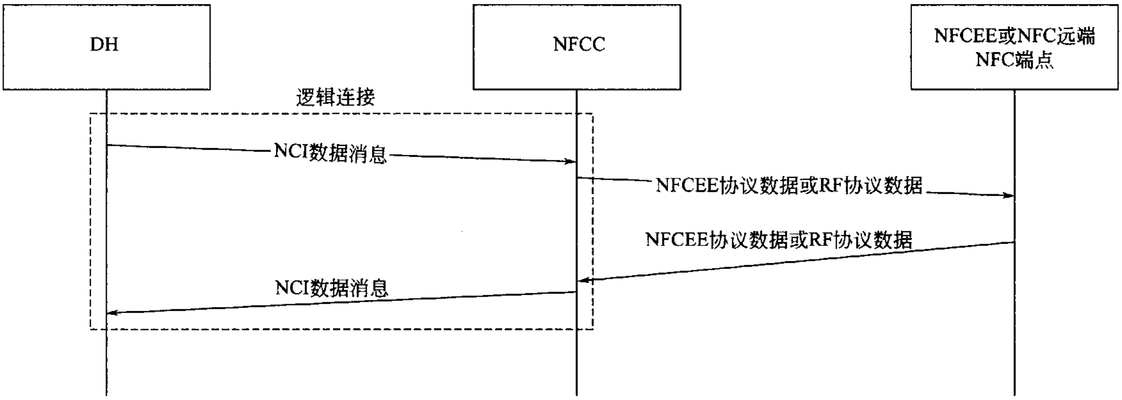


图 10-8　NCI 数据通信流程

据消息头中的连接 ID 来判断该数据的目的地及所采用的 RF 协议或 NFCEE 协议，然后将 NCI 数消息转换为 RF 协议规定的数据格式或 NFCEE 协议规定的数据格式向远端 NFC 端点或 NFCEE 发送。NCI 数据消息的流控基于 Credit 机制，前面已经介绍过。

NCI 数据消息格式如图 10-9 所示。连接 ID 用于表示该数据消息所属的逻辑连接。负载域用于承载上层协议的数据。

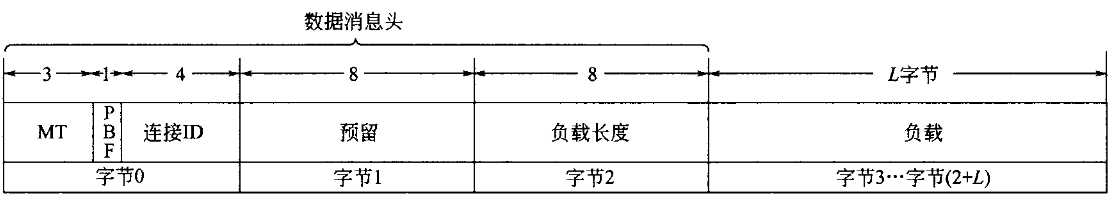


图 10-9　NCI 数据消息格式

10.3.5　消息的分段与重组

NCI 传输映射层为了将 NCI 消息适配进不同的物理传输协议（如 UART、SPI 等），定义了 MTU（Maximum Transmission Unit，最大传输单元）。当 DH 和 NFCC 之间要交互大于 MTU 的消息，包括控制消息和数据消息时，发送方（DH 或 NFCC）需要对消息分段，接收方（NFCC 或 DH）需要对分段的消息进行重组。因此，DH 和 NFCC 都要支持分段和重组功能。

控制消息的分段需要遵守以下规则：

① 分段后的控制消息应当保持一致的 MT、GID 和 OID。

② 从 DH 到 NFCC 的控制消息分段，除了最后一个分段消息，其他分段消息的负载长度采用最大控制消息负载长度所定义的值。

③ 从 NFCC 到 DH 的控制消息分段，无论负载长度是多少，一般会根据内部优化的目的，尽量将控制消息分成更小的段。

数据消息的分段需要遵守以下规则：

① 分段后的数据消息应当保持一致的 MT 和连接 ID。

② 若从 DH 到 NFCC 的数据消息长度超过最大数据消息负载长度所定义的值，就必须分段。

③ 从 NFCC 到 DH 的数据消息，无论负载长度是多少，出于传输缓存的优化目的，可以尽量将数据消息分成更小的段。

10.4 RF 发现过程

10.4.1 RF 发现的参数配置

图 10-10 总结了从重置 NFCC 到 RF 通信的大致流程。当 NFCC 上电启动或 NFCC 被开启时，DH 向 NFCC 发送 CORE _ RESET _ CMD 命令对 NFCC 重置；紧接着，DH 向 NFCC 发送初始化命令 CORE _ INIT _ CMD，对 NFCC 初始化；之后，DH 对 NFCC 进行参数配置，如轮询模式或侦听模式下的某些不具有默认值的参数。NFCC 被配置后，DH 向 NFCC 发送 RF 发现命令 RF _ DISCOVER _ CMD，NFCC 将执行 RF 发现过程，获取远端 NFC 端点的信息，并为每一个远端 NFC 端点分配唯一的 RF 发现 ID。当远端 NFC 端点支持多个 RF 协议时，NFCC 将其看成不同的远端 NFC 端点，根据 RF 协议分配不同的 RF 发现 ID。NFCC 把发现的远端 NFC 端点通告给 DH，DH 向 NFCC 发送选择命令 RF _ DISCOVER _ SELECT _ CMD 来选择某一个远端 NFC 端点。NFCC 收到选择命令后，与远端 NFC 端点进行设备激活过程，详细过程见第 4 章的介绍。NFCC 向 DH 发送接口激活通告消息 RF _ INTF _ ACTIVATED _ NTF，DH 收到该消息后，与 NFCC 建立逻辑连接，并开始与远端 NFC 端点进行数据交互。

图 10-10 中有一点需要说明，对于 Frame RF 接口来说，先执行 RF 接口激活，然后执行设备激活。这与其他 RF 接口是不同的，具体内容在 10.5 节介绍。

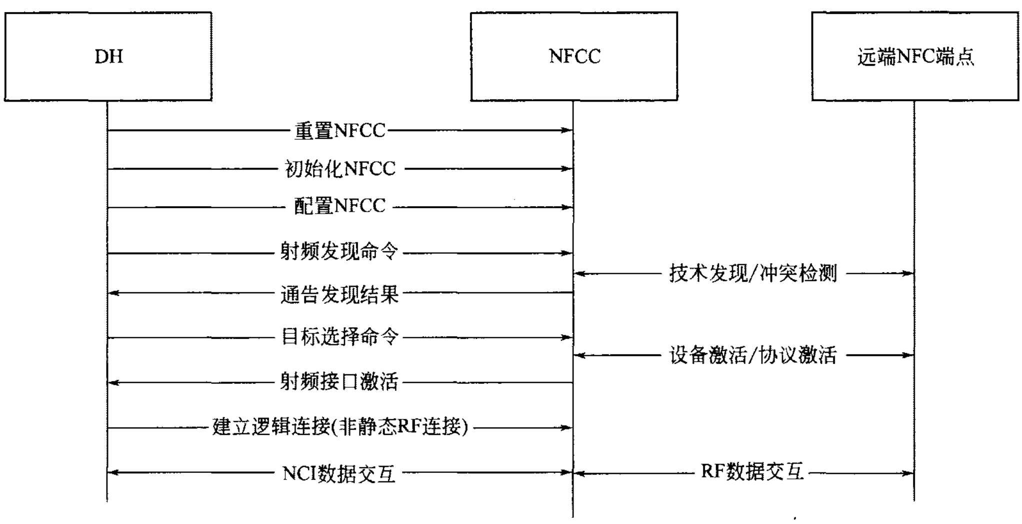


图 10-10 RF 通信流程

在 NFCC 执行 RF 发现过程之前，DH 需要对 NFCC 做以下配置：

① 轮询模式或侦听模式的参数。

② RF 接口与 RF 协议的映射表。

③ 侦听模式路由表。

轮询模式或侦听模式的参数与 RF 发现过程有关，通过 CORE _ SET _ CONFIG _ CMD

配置，如表 10-55 所示。这些参数用于不同的模式、技术及传输协议。如轮询模式 NFC-A 的参数，轮询模式 ISO-DEP 的参数，侦听模式 NFC-A 的参数等等。第 4 章介绍过，NFC 设备间或 NFC 设备与 NFC 标签之间需要技术发现、冲突检测、设备激活过程，这些过程中的命令和响应消息都携带参数。在 Activity 规范中，各个流程（如技术发现、冲突检测等）都有判决条件，如技术检测过程中完成了对 NFC-A 的检测后，通过判断条件参数来决定是否继续对 NFC-B 技术进行检测。这些参数中某些有默认值，某些则需要 DH 通过 CORE _ SET _ CONFIG _ CMD 配置。

DH 向 NFCC 发送 RF _ DISCOVER _ MAP _ CMD 命令来配置 RF 接口与 RF 协议之间的映射关系，如表 10-18 所示。RF 接口和 RF 协议映射表的作用是 NFCC 根据选择的 RF 协议来激活所对应的 RF 接口。RF _ DISCOVER _ MAP _ CMD 中的映射配置包括三个参数，即 RF 接口、RF 协议及模式。模式是指 RF 接口和 RF 协议之间的映射关系应用于侦听模式还是轮询模式。

表 10-18 RF _ DISCOVER _ MAP _ CMD

负载域	长度	值/描述		
映射配置的个数	1 字节	映射配置的个数。一组映射配置包括模式、RF 接口和 RF 协议		
映射配置[1…n]	3 字节	RF 协议	1 字节	见表 10-18
		模式	1 字节	b1＝1，侦听模式下，RF 接口与协议的映射 b0＝1，轮询模式下，RF 接口与协议的映射
		RF 接口	1 字节	见表 10-19

RF 协议如表 10-19 所示。

表 10-19 RF 协议

RF 协议取值	定义	RF 协议取值	定义
0x00	PROTOCOL_UNDETERMINED	0x05	PROTOCOL_NFC_DEP
0x01	PROTOCOL_T1T	0x06～0x7F	预留
0x02	PROTOCOL_T2T	0x80～0xFE	私有
0x03	PROTOCOL_T3T	0xFF	预留
0x04	PROTOCOL_ISO_DEP		

RF 接口如表 10-20 所示。NFCEE Direct RF 接口实际上是一个伪接口（Pseudo Interface）。当 NFCC 将数据路由至 NFCEE 时，NFCC 向 DH 发送 RF _ INTF _ ACTIVATED _ NTF 来激活该接口。NFCEE Direct RF 接口不与任何 RF 协议映射。ISO-DEP 协议与 ISO-DEP RF 接口映射，NFC-DEP 协议与 NFC-DEP RF 接口映射，T1T/T2T/T3T 协议与 Frame RF 接口映射。一般情况下，未包含在 RF _ DISCOVER _ MAP _ CMD 中的 RF 协议，默认都会映射到 Frame RF 接口。一个 RF 协议只能映射至一个 RF 接口。举例来说，T1T 协议与 Frame RF 接口映射的同时，不能再与其他 RF 接口映射。但是一个 RF 接口可以和多个 RF 协议映射，如 T1T、T2T、T3T 都可以同时与 Frame RF 接口映射。关于 RF 接口的详细内容，将在 10.5 节介绍。

NFCC 收到 RF _ DISCOVER _ MAP _ CMD 后，向 DH 响应 RF _ DISCOVER _ MAP _ RSP 消息，如表 10-21 所示。

表 10-20 RF 接口

RF 接口取值	定　义	RF 接口取值	定　义
0x00	NFCEE Direct RF 接口	0x04～0x7F	预留
0x01	Frame RF 接口	0x80～0xFE	私有
0x02	ISO-DEP RF 接口	0xFF	预留
0x03	NFC-DEP RF 接口		

表 10-21 RF _ DISCOVER _ MAP _ RSP

负　载　域	长　度	值/描述
状态	1 字节	见表 10-56

DH 通过 RF _ SET _ LISTEN _ MODE _ ROUTING _ CMD 命令向 NFCC 配置侦听模式路由表。侦听模式路由表的配置是可选的，作为发现过程的一部分。当 DH 设置 NFCC 工作在侦听模式下，并且 NFCC 支持侦听路由表时，DH 需要配置侦听模式路由表。侦听模式路由表的主要目的是告诉 NFCC 如何路由收到的数据。

NFCC 作为一个共享的资源，可以供 DH 和 NFCEE 使用。DH 通过 NFCC 与远端 NFC 端点通信；而 NFCEE 作为另一个主机，也可以通过 NFCC 与远端 NFC 端点通信。NFCC 工作在侦听模式下，从远端 NFC 端点收到数据后，无法判断该数据应该向哪个主机发送，因此需要侦听模式路由表提供路由信息。

侦听模式路由表中包含三种类型的路由条件，即基于 AID 的路由、基于协议的路由和基于技术的路由。基于 AID 的路由方式一般用于 NFCC 使用 ISO-DEP 通信协议的情况，并且至少能够解析 ISO 7816-4 中定义的 SELECT 命令。NFCC 收到 SELECT 命令后，通过解析该命令获取其中的 AID。并匹配路由表中的 AID。如果找到相同的 AID，将该命令转发至 AID 对应的 NFCEE ID。基于协议的路由指将 RF 协议作为路由条件，基于技术的路由将技术（NFC-A/B/F）作为路由的条件。

除了三种路由条件外，侦听模式路由表中还将电源状态作为路由依据。NCI 规范中定义了三种电源状态：馈电状态（Battery Off）、关机状态（Switched Off）和开机状态（Switch on）。馈电状态是指 NFC 设备完全没有电源供给，内部电池和外部电源都不可用。如 NFC 设备的电池被移除，DH 处于关机状态，此时，NFCC 从远端 NFC 设备产生的射频场中获取能量。关机状态是指 NFC 设备关机，即 DH 处于关闭状态，但是存在电源供给，如电池等。开机状态即 NFC 设备工作在正常开机状态。这里需要说明，在很多设计中，NFC 芯片与电池电源输出相连，因此在 DH 关闭的情况下，NFC 芯片也能获取电源供给。

RF _ SET _ LISTEN _ MODE _ ROUTING _ CMD 的格式如表 10-22 所示。一个路由 Entry 代表一种路由方式，每个路由方式包含 NFCEE ID、电源状态及路由条件。工作在侦听模式下的 NFCC 收到数据后，根据侦听模式路由表的路由方式将数据路由至指定的 NFCEE。如果没有找到匹配的路由 Entry，NFCC 将把数据发送给 DH。举一个简单的例子，DH 向 NFCC 配置了侦听模式路由表，指定在关机状态下，所有 ISO-DEP 协议的数据都路由到 NFCEE。当 NFC 设备（DH）关机后，NFCC 工作在侦听模式下（卡模拟模式），收到远端 NFC 设备发来的 SELECT 命令，NFCC 通过判断路由 Entry 后，将该命令路由至 NFCEE。

表 10-22 RF _ SET _ LISTEN _ MODE _ ROUTING _ CMD

负载域	长度	值/描述		
More	1 字节	0x00:最后一个消息 0x01:还有更多的消息 0x02～0xFF:预留		
路由 Entry 的个数	1 字节	每一个路由 Entry 都包括三个部分:类型、长度和值。路由 Entry 实际上就是路由的方式		
路由 Entry[1…n]	x+2 字节	类型	1 字节	0x00:表示基于技术路由方式 0x01:表示基于协议路由方式 0x02:表示基于 AID 路由方式 0x03～0x9F:预留 0xA0～0xFF:私有
		长度	1 字节	值的长度
		值	x 字节	如果是基于技术的路由方式,值包括 3 个字节: 字节 1:NFCEE ID 字节 2:电源状态 字节 3:技术 如果是基于协议的路由方式,值包括 3 个字节: 字节 1:NFCEE ID 字节 2:电源状态 字节 3:协议 如果是基于 AID 的路由方式,值包括 3 个字节: 字节 1:NFCEE ID 字节 2:电源状态 字节 3:AID

NFCC 收到 RF _ SET _ LISTEN _ MODE _ ROUTING _ CMD 后，将向 DH 响应RF _ SET _ LISTEN _ MODE _ ROUTING _ RSP 消息，如表 10-23 所示。

表 10-23 RF _ SET _ LISTEN _ MODE _ ROUTING _ RSP

负载域	长度	值/描述
状态	1 字节	见表 10-56

10.4.2 开始 RF 发现过程

DH 配置完 NFCC 后，就可以向 NFCC 发送 RF 发现命令来执行 RF 发现过程。RF 发现过程是一个周期执行轮询和侦听的活动。在对 NFCC 的配置中，参数 TOTAL _ DURATION 用于表示一个 RF 发现周期的长度，用毫秒（ms）表示，参见表 10-55。该时间参数只是为 NFCC 提供大致的值，NFCC 根据具体情况进行调整。

通过 CORE _ SET _ CONFIG _ CMD 配置的与 RF 发现过程相关的参数没有指定 NFCC 应当工作在侦听还是轮询模式下，DH 通过 RF _ DISCOVER _ CMD 命令使 NFCC 执行 RF 发现过程。在该命令中指定了 NFCC 应当工作在什么射频技术与模式下，以及该射频技术与模式的执行时间。NCI 规范支持的射频技术与模式如表 10-24 所示。射频技术与模式包含三个部分：技术、工作模式（轮询/侦听）和通信模式（主动/被动）。由于目前 Digital 1.0 规范不支持主动通信模式，因此在射频技术与模式列表中，凡是与主动通信相关的取值都标

注了“预留”。另外，由于 NFC Forum 正式发布的规范不支持 ISO 15693，所以与 ISO 15693 相关的取值也标注了“预留”。

表 10-24 射频技术与模式

值	描 述	值	描 述
0x00	NFC_A_PASSIVE_POLL_MODE	0x80	NFC_A_PASSIVE_LISTEN_MODE
0x01	NFC_B_PASSIVE_POLL_MODE	0x81	NFC_B_PASSIVE_LISTEN_MODE
0x02	NFC_F_PASSIVE_POLL_MODE	0x82	NFC_F_PASSIVE_LISTEN_MODE
0x03	NFC_A_ACTIVE_POLL_MODE(预留)	0x83	NFC_A_ACTIVE_LISTEN_MODE(预留)
0x04	预留	0x84	预留
0x05	NFC_F_ACTIVE_POLL_MODE(预留)	0x85	NFC_F_ACTIVE_LISTEN_MODE(预留)
0x06	NFC_15693_PASSIVE_POLL_MODE(预留)	0x86	NFC_15693_PASSIVE_LISTEN_MODE(预留)
0x07～0x6F	预留	0x87～0xEF	预留
0x70～0x7F	私有	0xF0～0xFF	私有

RF _ DISCOVER _ CMD 如表 10-25 所示。其中，每组配置都包含两个部分：射频技术与模式以及发现频率。发现频率的配置取决于 CORE _ INIT _ RSP 消息中 NFCC 特性 b0 的取值，如表 10-6 所示。b0=1，表示 RF _ DISCOVER _ CMD 支持 RF 发现频率的设置；如果b0=0，表示不支持。

表 10-25 RF _ DISCOVER _ CMD

<table>
<tr><th>负 载 域</th><th>长 度</th><th colspan="4">值/描述</th></tr>
<tr><td>配置的个数</td><td>1 字节</td><td colspan="4">配置的个数。一个配置包括两个部分:RF 技术与模式,发现频率</td></tr>
<tr><td rowspan="5">配置[0...n]</td><td rowspan="5">2 字节</td><td>RF 技术与模式</td><td>1 字节</td><td colspan="2">见表 10-24</td></tr>
<tr><td rowspan="4">发现频率</td><td rowspan="4">1 字节</td><td>0x00</td><td>预留</td></tr>
<tr><td>0x01</td><td>表示每一个发现周期中都会执行该射频技术与模式</td></tr>
<tr><td>0x02～0x0A</td><td>该范围内的取值只用于轮询模式,定义了特定技术的轮询执行的频率。举例来说,如果取值为 10,表示每 10 次发现周期执行一次该技术的轮询</td></tr>
<tr><td>0x0B～0xFF</td><td>预留</td></tr>
</table>

RF _ DISCOVER _ CMD 命令中可以携带多个不同的射频技术与模式。如果同时配置了轮询模式和侦听模式，那么侦听模式的持续时间为 TOTAL _ DURATION 减去轮询模式的时间。这里的轮询模式持续时间是指执行技术发现过程的时间。如果仅仅配置了轮询模式，空闲持续时间为 TOTAL _ DURATION 减去轮询模式持续时间。如果仅仅配置了侦听模式，TOTAL _ DURATION 将不起作用，被忽略。一般情况下，NFCC 会将不同技术的侦听合并在一起执行，同时把不同技术的轮询合并在一起执行。

NFCC 收到 RF _ DISCOVER _ CMD 后，将向 DH 响应 RF _ DISCOVER _ RSP，如表 10-26 所示。

表 10-26 RF _ DISCOVER _ RSP

负 载 域	长 度	值/描述
状态	1 字节	见表 10-56

NFCC 根据 RF _ DISCOVER _ CMD 中的射频技术与模式以及发现频率执行 RF 发现过程。RF 发现过程即第 4 章介绍的技术发现、冲突检测等过程。举例来说，如果射频技术与模式列表中包含 NFC _ A _ PASSIVE _ POLL _ MODE，NFCC 将发送 SENS _ REQ/ALL _ REQ 命令进行技术发现，通过 SDD _ REQ 进行冲突检测，识别出所有的侦听设备。

在轮询模式下，如果存在多个远端 NFC 端点或一个远端 NFC 端点支持多个 RF 协议，NFCC 将向 DH 发送多个 RF _ DISCOVER _ NTF 通告消息。需要注意的是，如果只发现了一个远端 NFC 端点，并且只支持一种 RF 协议，NFCC 将不会向 DH 发送 RF _ DISCOVER _ NTF，而是根据 RF 协议向 DH 发送 RF 接口激活的通告消息。RF _ DISCOVER _ NTF 如表 10-27 所示。

表 10-27 RF _ DISCOVER _ NTF

负 载 域	长 度	值/描述	
RF 发现 ID	1 字节	0：预留 1～254：由 NFCC 动态分配 255：预留	
RF 协议	1 字节	远端 NFC 端点支持的 RF 协议，见表 10-19	
射频技术与模式	1 字节	见表 10-24	
射频技术相关参数的长度	1 字节	射频技术相关参数的长度	
射频技术相关参数	0～n 字节	见表 10-28～表 10-33	
通知类型	1 字节	0	最后一个通告消息
		1	最后一个通告消息（由于 NFCC 的资源限制）
		2	还有更多的通告消息
		3～255	预留

NFCC 将为每一个远端 NFC 端点分配唯一的 RF 发现 ID。如果一个远端 NFC 端点支持多个 RF 协议，那么 NFCC 根据不同的 RF 协议产生多个 RF _ DISCOVER _ NTF。对于在同一个轮询周期内发现的支持多个 RF 协议的远端 NFC 端点，NFCC 将为其分配一致的 RF 发现 ID；如果不是在同一个周期内发现的支持多个 RF 协议的远端 NFC 端点，NFCC 将分配不同的 RF 发现 ID。RF 发现 ID 和 RF 协议的组合能够唯一地表示一个 RF _ DISCOVER _ NTF 消息。

由于可能存在多个远端 NFC 端点或一个远端 NFC 端点支持多个 RF 协议的情况，因此 NFCC 会向 DH 发送多个 RF _ DISCOVER _ NTF 消息。通知类型表示 NFCC 是否已经完成所有远端 NFC 端点的冲突检测。如果通知类型设置为 0 或 1，那么当前的 RF _ DISCOVER _ NTF 消息为最后一个通告消息，表明 NFCC 已经完成冲突检测。

RF _ DISCOVER _ NTF 通告消息中包含远端 NFC 端点支持射频技术的相关参数。这些参数实际上就是在技术发现、冲突检测过程中从远端 NFC 端点获取的。射频技术相关参数见表 10-28～表 10-33。具体来说，对于处于轮询模式下的 NFCC，它将收到的 SENS _ RES 和 SEL _ RES、SENSB _ RES、SENSF _ RES 消息作为射频技术相关参数封装在RF _ DISCOVER _ NTF 消息中，并发送给 DH。

表 10-28 轮询模式 NFC-A 参数

参　数	长　度	描　述
SENS_RES	2 字节	见第 4 章
NFCID1 长度	1 字节	见第 4 章
NFCID1	4、7 或 10 字节	见第 4 章
SEL_RES 长度	1 字节	SEL_RES 的长度
SEL_RES	0、1 字节	见第 4 章

表 10-29 侦听模式的 NFC-A 参数

参　数	长　度	描　述
暂时无参数		

表 10-30 轮询模式的 NFC-B 参数

参　数	长　度	描　述
SENSB_RES 长度	1 字节	见第 4 章
SENSB_RES	11 或 12 字节	见第 4 章

表 10-31 侦听模式的 NFC-B 参数

参　数	长　度	描　述
暂时无参数		

表 10-32 轮询模式的 NFC-F 参数

参　数	长　度	描　述	
比特率	1 字节	1	212Kbps
		2	424Kbps
		0 和 3～255	预留
SENSF_RES 长度	1 字节	见第 4 章	
SENSF_RES	16 或 18 字节	见第 4 章	

表 10-33 侦听模式的 NFC-F 参数

参　数	长　度	描　述
NFCID2 长度	1 字节	见第 4 章
NFCID2	0 或 8 字节	见第 4 章

对于处于侦听模式下并支持 NFC-A 或 NFC-B 技术的 NFCC 来说，收到 SENS_REQ/SEL_REQ 或 SENSB_REQ 后，会向 DH 发送 RF_DISCOVER_NTF，但是其中的射频技术相关参数为空（对于 NFC-B 来说，下一版 NCI 规范将把 SENSB_REQ 作为射频相关技术参数。对于 NFC-A 来说，SENS_REQ 本身不携带任何参数，因此无需作为射频技术相关参数）。对于处于侦听模式下支持 NFC-F 的 NFCC 来说，收到的 SENSF_REQ 中包含了 NFCID2，因此 NFCC 将 NFCID2 作为射频技术相关参数通过 RF_DISCOVER_NTF 消息发给 DH。

10.4.3 选择目标

在 DH 收到最后一个 RF_DISCOVER_NTF 后，DH 就可以选择远端 NFC 端点或者停止 RF 发现过程。DH 向 NFCC 发送 RF_DISCOVER_SELECT_CMD 来选择其中一个远端 NFC 端点进行通信。RF_DISCOVER_SELECT_CMD 如表 10-34 所示，其中指定了 RF 发现 ID、RF 协议以及 RF 接口。

表 10-34 RF_DISCOVER_SELECT_CMD

负载域	长度	值/描述
RF 发现 ID	1 字节	0:预留 1～254:由 NFCC 动态分配 255:预留
RF 协议	1 字节	见表 10-19
RF 接口	1 字节	见表 10-20

NFCC 收到 RF_DISCOVER_SELECT_CMD 后，向 DH 响应 RF_DISCOVER_SELECT_RSP，如表 10-35 所示。

表 10-35 RF_DISCOVER_SELECT_RSP

负载域	长度	值/描述
状态	1 字节	见表 10-56

如果 RF 发现 ID、RF 协议、RF 接口无效，状态参数将被设置为 STATUS_REJECT，否则状态参数设置为 STATUS_OK。NFCC 执行设备的激活过程，如第 4 章所述。设备激活完毕后，NFCC 将执行 RF 接口的激活操作。

需要注意的是，正如前面所提到的，如果在 RF 发现过程中只发现了一个远端 NFC 端点，并且只支持一种 RF 协议，NFCC 将不会发送 RF_DISCOVER_NTF 消息，DH 也就不会选择。NFCC 完成设备的激活过程，然后根据 RF 接口映射表激活 RF 接口。

NCI 提供了目标选择命令，但是没有规定选择的依据。在 Android 代码实现中，首先选择支持 P2P 功能的设备，即支持 NFC-DEP 协议的设备。如果检测出多个支持 NFC-DEP 协议的设备，会优选支持 NFC-F 技术的设备。如果没有 NFC-F，或存在多个支持 NFC-F 技术的设备，会选择第一个被发现的设备。如果没有支持 P2P 功能的设备，会选择第一个被发现的标签。从 Android 代码的实现来看，如果 NFC 设备周围存在一个标签和一个支持 P2P 功能的 NFC 设备，将优先选择 P2P 功能的目标设备进行通信。如果存在多个 P2P 功能的目标设备，优先选择支持 NFC-F 技术的目标设备，因为 NFC-F 的通信速率比 NFC-A 和 NFC-B 快。对于标签的选择，完全是按照发现的顺序进行选择，即选择第一个被发现的标签。

10.4.4 RF 接口激活与去激活

RF_INTF_ACTIVATED_NTF 用于通告 DH 相应的 RF 接口被激活，如表 10-36 所示。NFCC 与远端 NFC 端点完成设备激活后，根据 RF 接口映射表中对应关系，向 DH 发送射频接口激活通知消息。从表 10-36 可以看出，RF 接口激活过程中采用的射频技术与模式，数据的收发比特率与在数据交换过程中采用的射频技术与模式，数据的收发比特率可以不同，正如第 4 章介绍的，在 NFC-DEP 传输协议中，可以通过 PSL_REQ/PSL_RES 协

商数据交换过程中的参数。如果在设备/协议激活过程中使用 PSL _ REQ/PSL _ RES 协商参数，RF _ INTF _ ACTIVATED _ NTF 将协商后的结果通知给 DH。

RF _ INTF _ ACTIVATED _ NTF 中的激活参数是指在设备激活过程中，NFCC 收到的 ATTRIB 命令、ATTRIB 响应、RATS 命令、RATS 响应、ATR _ REQ 及 ATR _ RES。

表 10-36 RF _ INTF _ ACTIVATED _ NTF

负载域	长度	值/描述
RF 发现 ID	1字节	0:预留 1～254:由 NFCC 动态分配 255:预留
RF 接口	1字节	见表 10-20
RF 协议	1字节	见表 10-19
激活的射频技术与模式	1字节	见表 10-24,目前正在使用的射频技术和模式
最大数据负载长度	1字节	1～255
初始的 Credit 值	1字节	取值范围 0x00～0xFE
射频技术相关的参数长度	1字节	射频技术相关的参数长度
射频技术相关的参数	0～n 字节	见表 10-28～表 10-33
数据交换所采用的射频技术与模式	1字节	在数据交换过程中采用的射频技术和模式
数据交换时的传输比特率	1字节	见表 10-37
数据交换时的接收比特率	1字节	见表 10-37
激活参数的长度	1字节	激活参数的长度
激活参数	0～n 字节	对于 ISO-DEP 来说,参数可以是 ATTRIB 命令、ATTRIB 响应、RATS 命令、RATS 响应。对于 NFC-DEP 来说,参数可以是 ATR_REQ 或 ATR_RES

数据比特率如表 10-37 所示。

表 10-37 数据比特率

比特率	定义	比特率	定义
0x00	NFC_BIT_RATE_106:106 Kbps	0x05	NFC_BIT_RATE_3390:3390 Kbps
0x01	NFC_BIT_RATE_212:212 Kbps	0x06	NFC_BIT_RATE_6780:6780 Kbps
0x02	NFC_BIT_RATE_424:424 Kbps	0x07～0x7F	预留
0x03	NFC_BIT_RATE_848:848 Kbps	0x80～0xFE	私有
0x04	NFC_BIT_RATE_1695:1695 Kbps	0xFF	预留

在 NFCC 发送 RF _ INTF _ ACTIVATED _ NTF 后，静态射频连接就可以使用了。RF _ INTF _ ACTIVATED _ NTF 中的最大数据负载长度和初始 Credit 这两个参数可以应用到静态射频连接中。动态连接创建的过程参见 10.3.3.6 节。

当数据通信结束后，RF 接口需要被去激活（Deactivation）。RF 接口的去激活可以由 DH 发起，也可以由远端 NFC 端点发起。DH 向 NFCC 发送 RF _ DEACTIVATE _ CMD 来去激活 RF 接口，如表 10-38 所示。其中，去激活类型表示 NFCC 收到去激活命令后，将转至什么状态。RF 发现状态机将在 10.4.6 节介绍。

表 10-38 RF _ DEACTIVATE _ CMD

负载域	长度	值/描述
去激活类型	1 字节	0x00:Idle 0x01:Sleep 0x02:Sleep_AF 0x03:Discovery 0x04～0xFF:预留

NFCC 收到 RF _ DEACTIVATE _ CMD 后，将响应 RF _ DEACTIVATE _ RSP，如表 10-39 所示。

表 10-39 RF _ DEACTIVATE _ RSP

负载域	长度	值/描述
状态	1 字节	见表 10-56

NFCC 收到 RF _ DEACTIVATE _ CMD，向 DH 返回 RF _ DEACTIVATE _ RSP 之后，执行设备的去激活，如发送 DSL _ REQ/RLS _ REQ 消息。设备去激活后，NFCC 向 DH 通告 RF _ DEACTIVATE _ NTF，其中包含去激活的类型和原因，如表 10-40 所示。

表 10-40 RF _ DEACTIVATE _ NTF

负载域	长度	值/描述
去激活类型	1 字节	0x00:Idle 0x01:Sleep 0x02:Sleep_AF 0x03:Discovery 0x04～0xFF:预留
去激活原因	1 字节	0x00:DH_Request,由 DH 请求去激活 0x01:Endpoint_Request,由远端 NFC 端点请求去激活 0x02:RF_Link_Loss,RF 链路发生断开 0x03:NFC-B_Bad_AFI,NFC-B 中的 AFI 无法解析 0x04～0xFF:预留

10.4.5 NFCEE 的 RF 发现请求

除了 DH 可以向 NFCC 配置 RF 发现参数外，NFCEE 作为另一个主机也可以向 NFCC 配置 RF 发现的相关参数或发起射频发现请求。虽然 NFCEE 与 NFCC 之间的通信并不在 NCI 范围内，但是 NCI 定义了 RF _ NFCEE _ DISCOVERY _ REQ _ NTF 消息。当 NFCEE 修改了 NFCC 的 RF 发现参数（如 RF 协议、射频技术和模式）后，NFCC 向 DH 发送该通告消息，告诉 DH RF 发现参数被 NFCEE 修改，如表 10-41 所示。

表 10-41 RF _ NFCEE _ DISCOVERY _ REQ _ NTF

<table>
<tr><th>负载域</th><th>长度</th><th colspan="3">值/描述</th></tr>
<tr><td>信息 Entry 的个数</td><td>1 字节</td><td colspan="3">信息 Entry 的个数。每个 Entry 包含三个部分:类型、长度和值</td></tr>
<tr><td rowspan="3">信息 Entry[1…n]</td><td rowspan="3">x+2 字节</td><td>类型</td><td>1 字节</td><td>0x00:表示增加了 RF 发现请求参数
0x01:表示删除了 RF 发现请求参数
0x02～0x7F:预留
0x80～0xFF:私有</td></tr>
<tr><td>长度</td><td>1 字节</td><td>值的长度</td></tr>
<tr><td>值</td><td>x 字节</td><td>值包括 3 字节:
字节 1:NFCEE ID
字节 2:射频技术与模式
字节 3:RF 协议</td></tr>
</table>

此外，NCI定义了RF _ NFCEE _ ACTION _ NTF消息，允许NFCC向DH通告与NFCEE相关的动作，如表10-42所示。这里所说的动作，可以是判断路由，收到某个AID的SELECT命令等。举例来说，当NFCC收到SELECT命令后，发现其中包含的AID在侦听模式路由表中，于是将该SELECT命令转发至NFCEE；同时，NFCC向DH发送RF _ NFCEE _ ACTION _ NTF消息，告诉DH有一个SELECT命令被NFCC路由到NFCEE。这里的支持数据依赖于触发条件。举例来说，如果触发条件为AID，支持数据将包含该AID；如果触发条件为基于RF协议的路由方式，支持数据包含该RF协议。

表10-42 RF _ NFCEE _ ACTION _ NTF

负载域	长度	值/描述
NFCEE ID	1字节	NFCEE ID
触发条件	1字节	0x00:SELECT命令中的AID 0x01:基于RF协议的路由 0x02:基于RF技术的路由 0x03～0x0F:预留 0x10:基于AID 0x11～0xFF:预留
支持数据的长度	1字节	支持数据的长度
支持数据	*n*字节	依赖于触发条件

NFCC发送RF _ NFCEE _ ACTION _ NTF的功能需要DH对NFCC进行配置，即通过CORE _ SET _ CONFIG _ CMD命令配置RF _ NFCEE _ ACTION参数，如表10-43所示。

表10-43 RF _ NFCEE _ ACTION参数

ID	长度	值	描述
RF_NFCEE_ACTION	1Octet	0x00	NFCC不会发送RF_NFCEE_ACTION_NTF消息
		0x01(默认)	NFCC发送RF_NFCEE_ACTION_NTF消息
		0x02～0xFF	预留

10.4.6 RF发现状态机

NCI的整个RF通信过程维护了一个状态机，如图10-11所示。状态机一共有7个状态，分别为RFST _ IDLE、RFST _ DISCOVERY、RFST _ W4 _ ALL _ DISCOVERIES、RFST _ W4 _ HOST _ SELECT、RFST _ W4 _ HOST _ SELECT、RFST _ POLL _ ACTIVE和RFST _ LISTEN _ ACTIVE。

当NFCC上电启动后，DH对NFCC完成初始化及相关配置，NFCC进入RFST _ IDLE状态。当DH向NFCC发送RF _ DISCOVER _ CMD后，NFCC进入RFST _ DISCOVERY状态。在该状态下，NFCC根据RF发现配置周期的执行侦听和轮询模式。当NFCC发现多个远端NFC端点或一个远端NFCC端点支持多个RF协议时，NFCC进入RFST _ W4 _ ALL _ DISCOVERIES状态。如果NFCC只发现了一个远端NFC端点且只支持一种RF协议，NFCC向DH发送RF _ INTF _ ACTIVATED _ NTF消息，并进入RFST _ POLL _ ACTIVE。在该状态下，处于轮询模式的NFCC被激活，可以进行RF通信。在RFST _ W4 _ ALL _ DISCOVERIES状态下，如果NFCC检测出所有的远端NFC端点，并向DH发送最后一个RF _ DISCOVER _ NTF，NFCC进入RFST _ W4 _ HOST _ SELECT状态。

在该状态下，NFCC 等待 DH 发送的目标选择命令 RF _ DISCOVER _ SELECT _ CMD。当 NFCC 收到 RF _ DISCOVER _ SELECT _ CMD，与远端 NFC 端点进行设备激活，并向 DH 发送 RF 接口激活命令 RF _ INTF _ ACTIVATED _ NTF，进入 RFST _ POLL _ ACITVE。

如果 NFCC 处于侦听模式下，收到远端 NFC 端点的设备激活命令后，向 DH 发送RF _ INTF _ ACTIVATED _ NTF 命令来激活 RF 接口，并进入 RFST _ LISTEN _ ACTIVE 状态。在该状态下，处于侦听模式的 NFCC 已经被激活，可以进行 RF 通信。

从图 10-11 可以看出，RF 接口的去激活使 NFCC 从一个状态跳转到指定的状态。

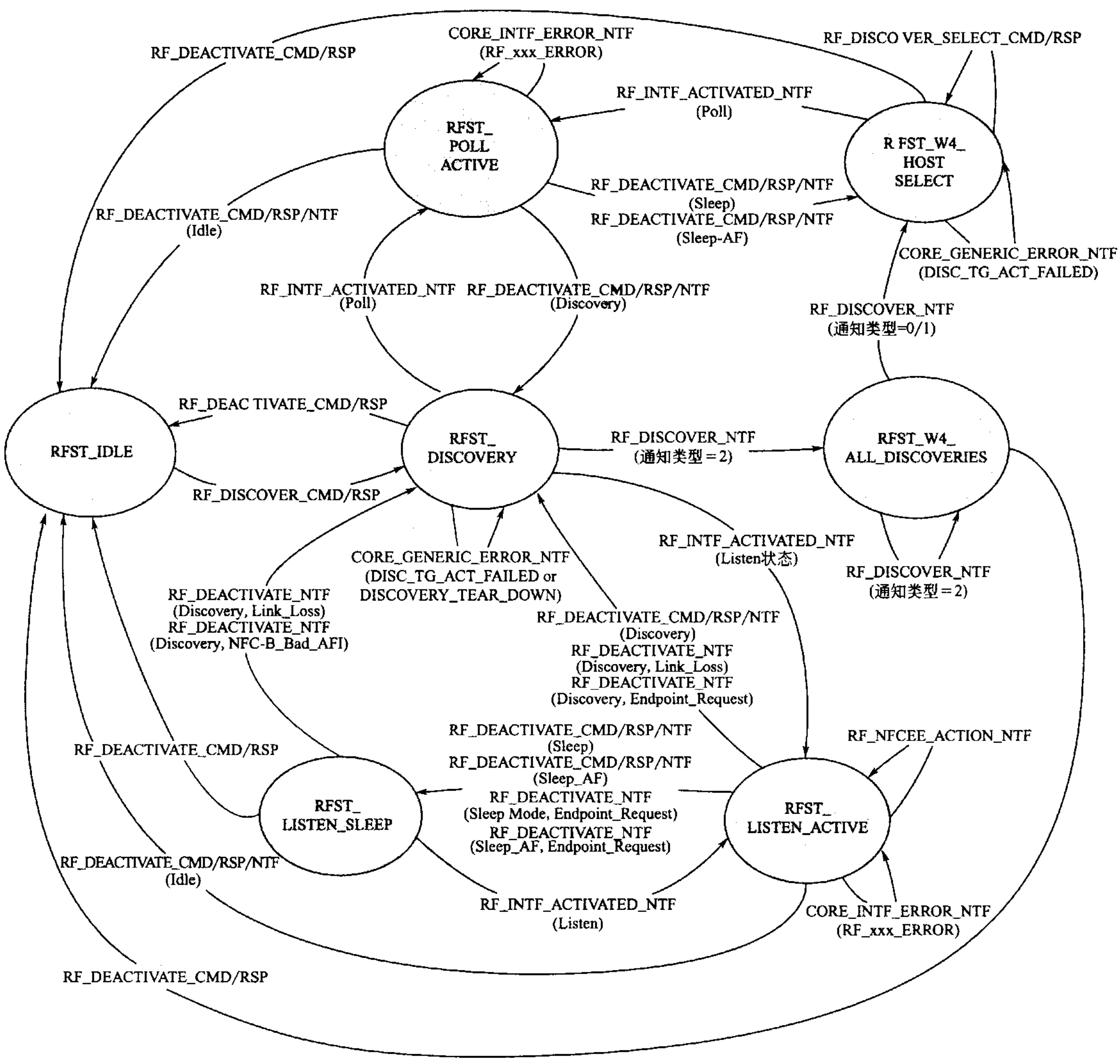


图 10-11 状态机

10.5 RF 接口

10.5.1 RF 接口架构

NFCC 与远端 NFCC 之间采用的通信协议称为 RF 协议。RF 协议可以是 NFC-DEP、

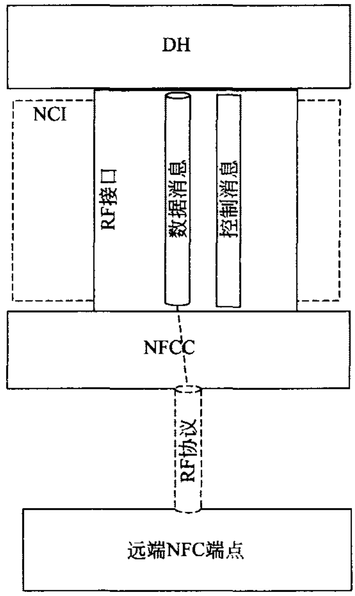


图 10-12 RF 协议与 RF 接口

ISO-DEP、T1T、T2T 和 T3T，具体如表 10-19 所示。RF 接口用于 DH 和远端 NFC 端点通信。不同的 RF 协议对应不同的 RF 接口。RF 接口根据 RF 协议分为 NFC-DEP RF 接口、ISO-DEP RF 接口、Frame RF 接口等。RF 接口与 RF 协议的关系如图 10-12 所示。从图中可以看出，对于 DH 来说，该 RF 协议没有运行在 DH 上，DH 无法“看到”该协议，因此 NCI 定义了 RF 接口的概念，DH 通过 RF 接口能够控制对应的 RF 协议。

除了以上 NCI 1.0 规范中提到的三种 RF 接口，实际上正在讨论的还有 LLCP High RF 接口、LLCP Low RF 接口、NDEF RF 接口等。这些接口将加入到 NCI 规范中。

为什么会有这么多 RF 接口呢？本质上还是 NFC 协议栈分割引起的。NFC 根据 3 种不同的工作模式有不同的协议栈架构，这些协议有些运行在 DH 上，有些运行在 NFCC 上。NFC 芯片（NFCC）有“胖”和“瘦”之分，“瘦”NFC 芯片的处理能力较弱，存储较小，因此只实现了 Analog 协议和部分 Digital 协议，完成 RF 数据到三种技术帧之间的转换功能；其他 NFC 协议，包括部分 Digital 协议，实际上都可以运行在 DH 之上。但是随着 NFC 芯片处理能力、存储能力的增强，NFC 芯片厂商为了增加 NFC 芯片的功能和竞争力，将 NFC 协议栈中的协议逐渐集成在 NFC 芯片中，也就是所说的“胖”NFC 芯片。NFC 的部分协议集成在 NFC 芯片中，从另一个角度来看，实际上减轻了 DH 的处理负担，对于操作系统提供商（如谷歌、微软等）来说，无需在操作系统中集成和维护相关的 NFC 协议栈。但是对于 NCI 中的 RF 接口来说，NFCC 上运行着不同的协议，DH 就需要有相应的 RF 接口与之对应，所以出现了众多的 RF 接口。下面根据不同的 RF 接口，介绍 NFC 协议分割。这里需要强调的是，在实际中由于 NFC 芯片厂商提供的 NFC 芯片所支持的协议各有不同，因此对于操作系统来说，只能提供尽可能全的协议栈支持。

读写模式下的协议分割与 RF 接口分类如图 10-13 所示。在读写模式下，如果 NFCC 实现 Digital 中的 Frame 和 NFC-A/B/F 的编码部分，DH 实现 ISO-DEP（即 ISO14443-4）及以上的协议部分，那么对于 DH 来说，DH 通过 NCI 能够“看到”的是 Digital 协议中的 Frame，因此 NCI 向 DH 提供 Frame RF 接口来控制 Digital 协议中的 Frame。如果 ISO-DEP 也实现在 NFCC 中即完整的 Digital 都实现在 NFCC 中，那么对于 DH 来说，通过 NCI 能够“看到”NFCC 上的 ISO-DEP，因此 NCI 向 DH 提供 ISO-DEP RF 接口来控制 ISO-DEP。如果 NDEF 相关的协议也实现在 NFCC 中，那么 DH 可以通过 NDEF RF 接口直接

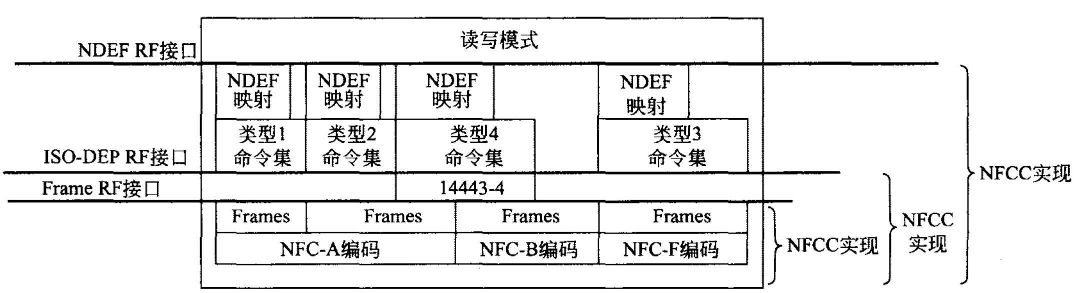


图 10-13 读写模式的 NFC 协议分割

交互 NDEF 消息。对于读写模式，NCI 1.0 规范目前只支持 ISO-DEP RF 接口和 Frame RF 接口。NDEF RF 接口将在后续的 NCI 版本中支持。关于 Digital 协议的架构，可以参考第 4 章的介绍。

卡模拟模式的协议分割和 RF 接口分类如图 10-14 所示。卡模拟模式与读写模式的协议分割和 RF 接口类似，包括 Frame RF 接口和 ISO-DEP RF 接口。

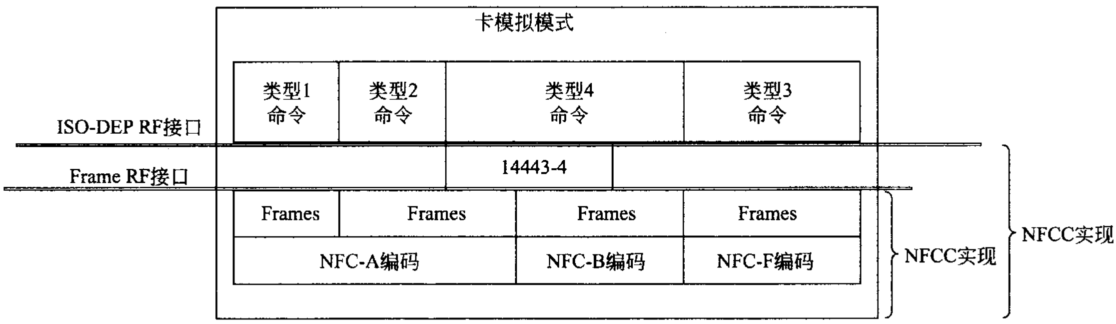


图 10-14 卡模拟模式的 NFC 协议分割

点对点模式的协议分割和 RF 接口分类如图 10-15 所示。点对点模式下，有 4 类 RF 接口：Frame RF 接口、NFC-DEP RF 接口、LLCP Low RF 接口和 LLCP High RF 接口。如果 NFCC 实现了 NFC-DEP 协议（ISO18092 NFCIP-1），那么 NCI 为 DH 提供 NFC-DEP RF 接口来控制 NFC-DEP 协议。在后续的 NCI 规范中，将加入 LLCP Low RF 接口和 LLCP High RF 接口。LLCP 协议分成两个部分，一部分包含 LLCP 链路维护功能 Symmetry 以及 LLCP 链路去激活功能，另一部分包括 LLCP 的其他功能。由于 LLCP Symmetry 功能需要周期地执行链路维护的例行工作，而且链路维护与链路去激活功能有一定的交集，因此将这两部分功能实现在 NFCC 中，让 NFCC 来完成"例行"工作，减少了 DH 的处理负担。NCI 提供 LLCP Low RF 接口供 DH 控制实现在 NFCC 中的 LLCP 部分协议。如果完整的 LLCP 协议都由 NFCC 实现，就有了 LLCP High RF 接口。

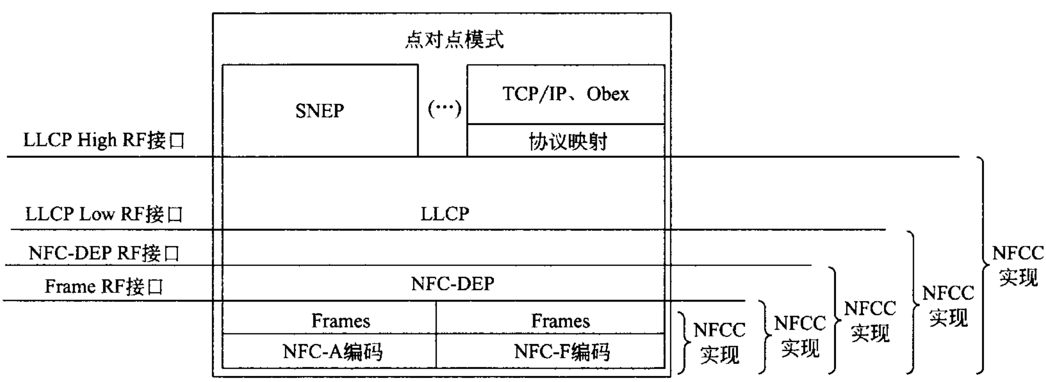


图 10-15 点对点模式的 NFC 协议分割

RF 接口提供了三个基本的功能：数据映射、RF 协议控制和 RF 接口管理。

数据映射是指将 NCI 数据消息映射进 RF 协议定义的数据格式中，或将 RF 协议的数据格式转换成 NCI 数据消息。举例来说，NFC-DEP RF 接口将 NCI 数据消息映射进 DEP_REQ/RES，并将 DEP_REQ/RES 消息映射进 NCI 数据消息。RF 协议控制是指 DH 通过

RF 接口向 NFCC 发送控制命令，如图 10-12 所示。控制消息主要用于配置或控制 RF 接口对应的 RF 协议。RF 接口管理主要完成接口的激活和去激活。下面将介绍 NCI 1.0 规范定义的 Frame RF 接口、ISO-DEP RF 接口、NFC-DEP RF 接口和 NFCEE RF 接口。

尽管 RF 接口的出现是由于 NFC 协议栈分割的原因，但是大部分情况下，NFCC 都会实现比较完整的 NFC 协议栈。NCI 为 DH 提供不同的 RF 接口，不同 RF 接口的存在并不会产生冲突。支持 Frame RF 接口，并不是说就不能支持 NFC-DEP RF 接口或 ISO-DEP RF 接口；支持了 NFC-DEP RF 接口，也不是说不能提供 Frame RF 接口。DH 和 NFCC 之间通过协商来获知各自能够支持的协议栈情况。Frame RF 接口可以说是一个最基本的 RF 接口，也是一个非常底层的接口，该接口能够直接控制三种技术的帧。在 RF_DISCOVER_MAP_CMD 中未包含的 RF 协议，都会默认映射至 Frame RF 接口。

10.5.2 Frame RF 接口

Frame RF 接口为 DH 提供了访问接口，能够访问 NFC 设备和远端 NFC 端点之间射频帧的负载。任何高层协议，如 ISO-DEP、NFC-DEP 等都需要由 DH 来处理，造成 RF 接口激活与设备激活是分离的。举例来说，ISO-DEP 协议中使用 RATS 或者 ATTRIB 进行设备激活，而 RATS 和 ATTRIB 由 DH 发送，因此当 RF 接口激活后，DH 才能与远端 NFC 端点执行设备激活过程。这一点与其他 RF 接口，如 NFC-DEP RF 接口和 ISO-DEP RF 接口是不同的。

DH 向 NFCC 发送 NCI 数据消息，NFCC 将 NCI 数据消息转换成 RF 帧发送给远端 NFC 端点。当 NFCC 从远端 NFC 端点收到 RF 帧之后，从射频帧中获取数据负载并转换为 NCI 数据消息发送给 DH。射频帧的格式取决于所采用的技术。图 10-16、图 10-17和图 10-18分别描述了 NCI 数据消息如何映射进 NFC-A、NFC-B 和 NFC-F 的射频帧。

图 10-19、图 10-20 和图 10-21 分别描述了 NFC-A、NFC-B 和 NFC-F RF 帧如何映射至 NCI 数据消息。

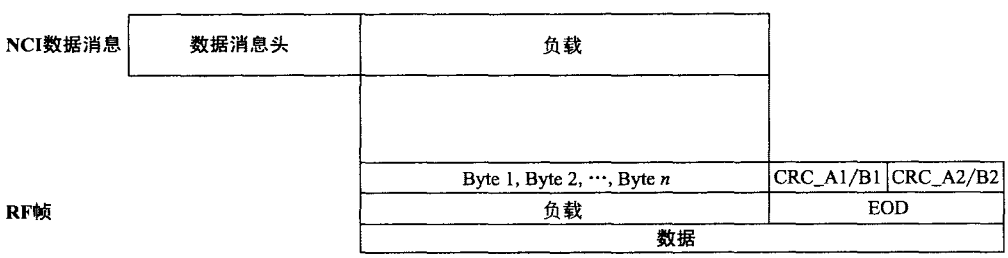


图 10-16 NCI 数据映射进 NFC-A RF 帧

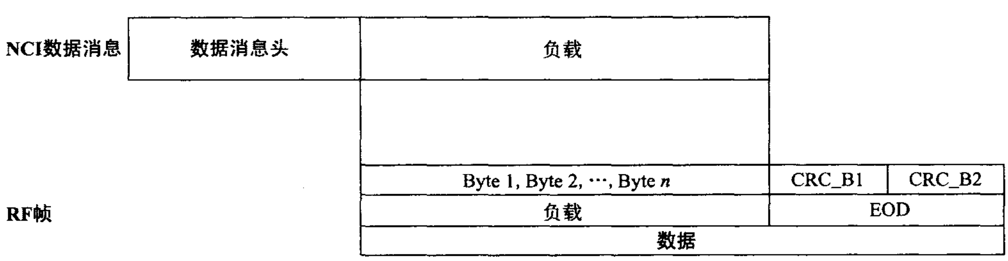


图 10-17 NCI 数据映射进 NFC-B RF 帧

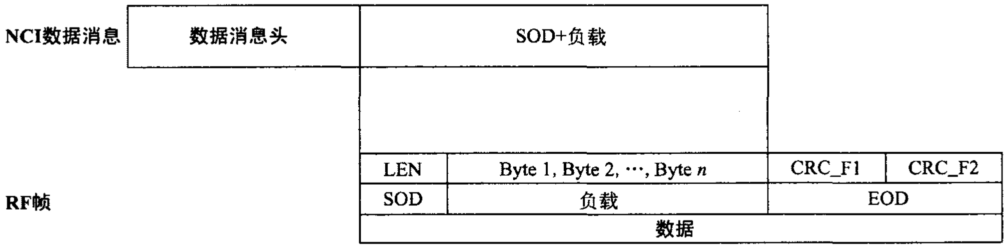


图 10-18 NCI 数据映射进 NFC-F RF 帧

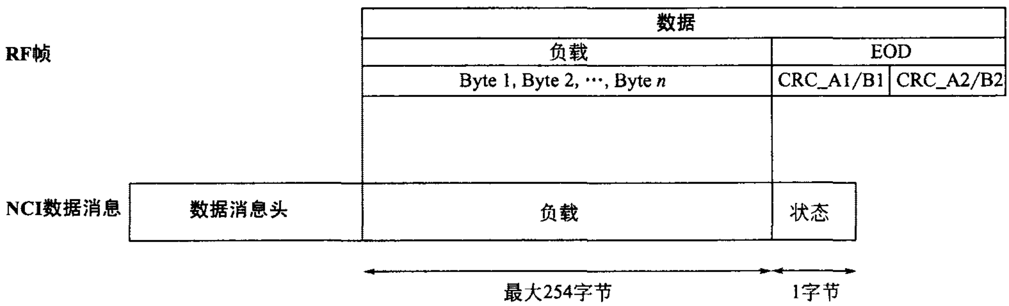


图 10-19 NFC-A RF 帧映射进 NCI 数据

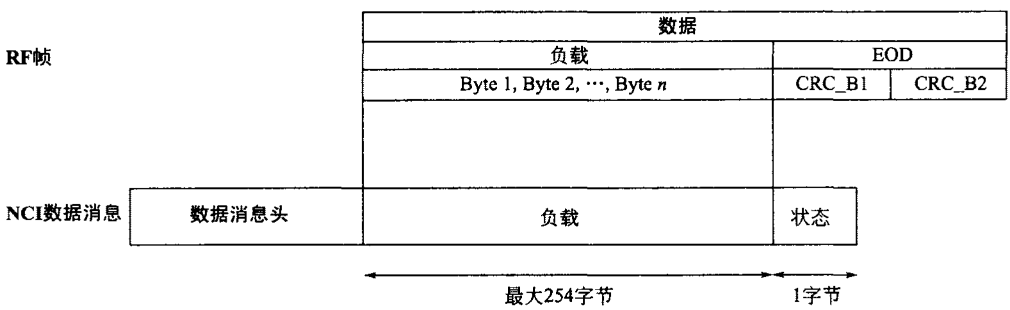


图 10-20 NFC-B RF 帧映射进 NCI 数据

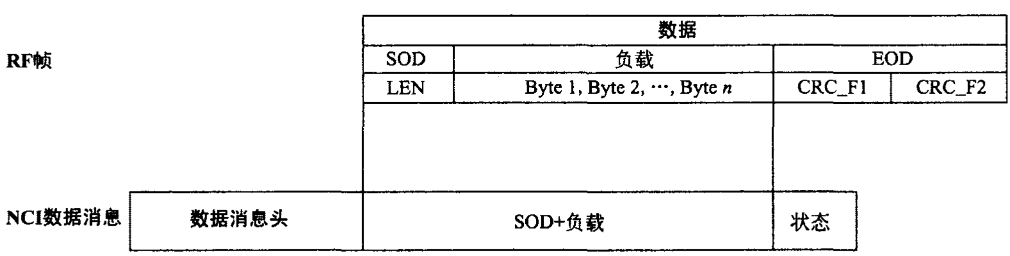


图 10-21 NFC-F RF 帧映射进 NCI 数据

当 Frame RF 接口激活后，DH 通过 Frame RF 接口修改射频通信的相关参数。DH 向 NFCC 发送 RF _ PARAMETER _ UPDATE _ CMD，修改相关射频通信参数，如表 10-44 所示。该命令只能在 RFST _ POLL _ ACTIVE 或 RFST _ LISTEN _ ACTIVE 状态下发送。

表 10-44 RF _ PARAMETER _ UPDATE _ CMD

负 载 域	长 度	值/描述		
参数的个数	1 字节	射频通信参数的个数		
射频通信参数[1…n]	x+2 字节	ID	1 字节	0x00:射频技术与模式,见表 10-24 0x01:发送比特率,见表 10-37 0x02:接收比特率,见表 10-37 0x03:NFC-B 数据交换配置,如 TR0、TR1 等 0x04～0x7F:预留 0x80～0xFF:私有
		长度	1 字节	值的长度
		值	x 字节	见 ID 对应的取值

NFCC 收到该命令后，向 DH 返回 RF _ PARAMETER _ UPDATE _ RSP，如表 10-45 所示。如果 NFCC 不支持某些射频参数，将状态参数设置为 STATUS _ INVALID _ PARAM，并返回无效参数的 ID。

表 10-45 RF _ PARAMETER _ UPDATE _ RSP

负 载 域	长 度	值/描述
状态	1 字节	见表 10-56
参数的个数	1 字节	参数的个数
射频通信参数 ID[0…n]	1 Octet	无效的参数 ID,见表 10-55

NFCC 发送响应消息后，与远端 NFC 端点进行参数协商。以 NFC-DEP 为例，将触发 PSL _ REQ/PSL _ RES 的交互。

对于支持 NFC-F 技术的 T3T RF 协议来说，除了能够在 RF 发现过程中发送 SENSF _ REQ 命令之外，还能够在 Frame RF 接口激活后，即在 RFST _ POLL _ ACTIVE 状态下发送 SENSF _ REQ 来发现处于侦听模式的 NFC-F 的远端 NFC 端点。对于 RF 协议 T3T 来说，DH 无法通过 Frame RF 接口使 NFCC 发送 SENSF _ REQ 命令，因此引入 RF _ T3T _ POLLING _ CMD。DH 通过 Frame RF 接口将 RF _ T3T _ POLLING _ CMD 发送给 NFCC。该命令携带了 SENSF _ REQ 中的参数，即字节 2～5，详见第 4 章的描述。NFCC 收到该命令后返回 RF _ T3T _ POLLING _ RSP，并向远端 NFC 端点发送 SENSF _ REQ 命令。当 NFCC 收到远端 NFC 端点的 SENSF _ RES 响应后，向 DH 发送 RF _ T3T _ POLLING _ NTF 通告消息，其中携带了收集到的所有 SENSF _ RES 消息。

10.5.3 ISO-DEP RF 接口

ISO-DEPRF 接口为 DH 提供了访问接口，能够访问 NFC 设备和远端 NFC 端点之间交互的 I-Blocks。任何高层协议，如 ISO 7816 APDU，都需要由 DH 处理。

DH 向 NFCC 发送 NCI 数据消息，NFCC 将 NCI 数据消息转换成一个或多个 I-Blocks。图 10-22 描述了 NCI 数据消息如何映射进 I-Blocks。

当 NFCC 收到远端 NFC 端点发送的 I-Blocks（NFCC 将收到的 RF 帧转换为 I-Blocks）后，将其映射进 NCI 数据消息，如图 10-23 所示。

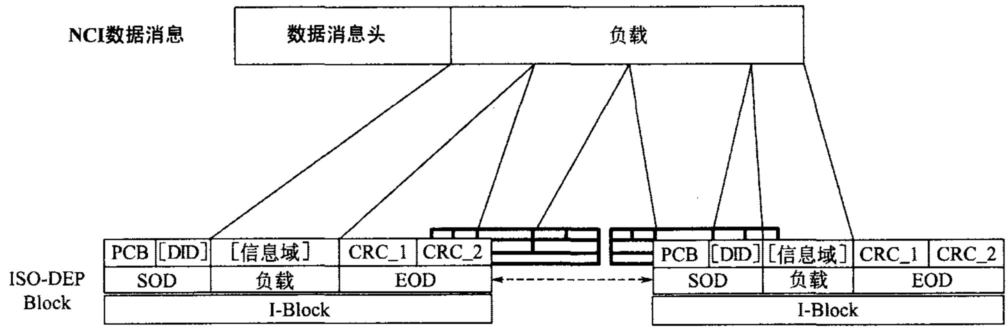


图 10-22　NCI 数据映射进 I-Blocks

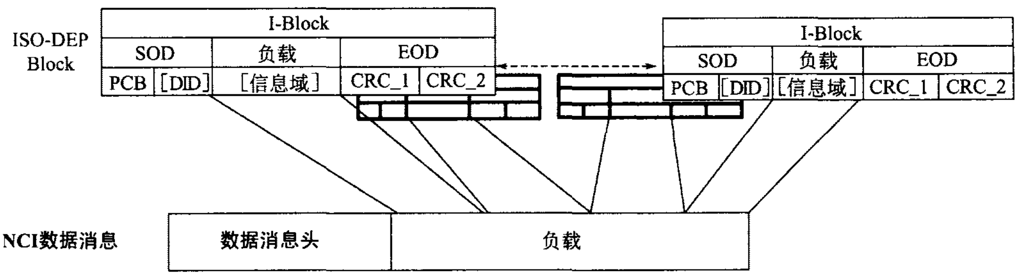


图 10-23　I-Blocks 映射进 NCI 数据

10.5.4 NFC-DEP RF 接口

NFC-DEPRF 接口为 DH 提供了访问接口，能够访问 NFC 设备和远端 NFC 端点之间交互的 NFC-DEP 数据帧（DEP _ REQ/DEP _ RES）。任何高层协议，如 LLCP，都需要由 DH 处理。

DH 向 NFCC 发送 NCI 数据消息，NFCC 将 NCI 数据消息转换成 DEP _ REQ（如果 NFCC 为发起方设备）或 DEP _ RES（如果 NFCC 为目标设备）。图 10-24 描述了 NCI 数据消息如何映射进 NFC-DEP 数据帧。

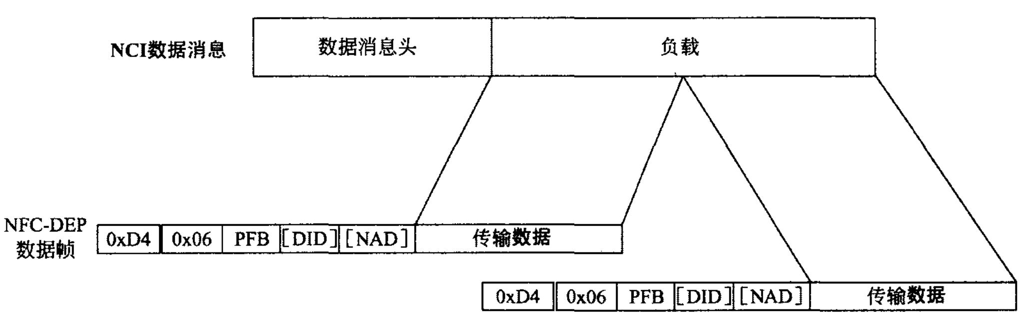


图 10-24　NCI 数据消息映射进 NFC-DEP 数据帧

当 NFCC 收到远端 NFC 端点发送的 NFC-DEP 数据帧后（NFCC 将收到的 RF 帧转换为 NFC-DEP 数据帧）后，将其映射进 NCI 数据消息，如图 10-25 所示。

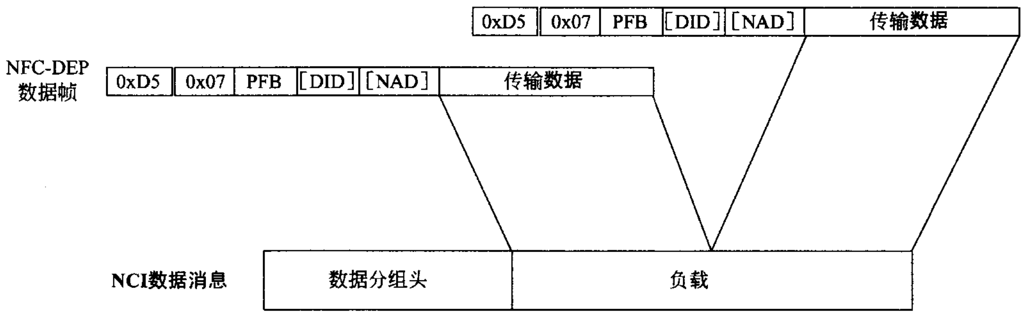


图 10-25 NFC-DEP 数据帧映射进 NCI 数据消息

10.5.5 NFCEE Direct RF 接口

NFCEE Direct RF 接口是一个“伪”接口。当 NFCC 从远端 NFC 端点收到的数据被路由至 NFCEE 时，DH 通过该接口获取相关信息，详见 10.4.5 节。NFCEE Direct RF 接口不与任何 RF 协议映射。当 NFCC 将数据路由至 NFCEE 时，NFCC 向 DH 发送 RF _ INTF _ ACTIVATED _ NTF 来激活该接口。

10.6 NFCEE 发现

10.6.1 NFCEE 发现过程

DH 向 NFCC 发送 NFCEE _ DISCOVER _ CMD 命令发现 NFCC 上连接的 NFCEE，如表 10-46 所示。NFCEE _ DISCOVER _ CMD 中携带了打开或关闭 NFCEE 发现的选项。当操作参数设置为 0x00 时，NFCC 不会向 DH 报告关于 NFCEE 的状态信息。如果操作参数设置为 0x01，当 NFCEE 从 NFCC 上连接或移除时，NFCC 将向 DH 发送 NFCEE _ DISCOVER _ NTF 进行通告。

表 10-46 NFCEE _ DISCOVER _ CMD

负 载 域	长 度	值/描述	
操作	1 字节	0x00	关闭 NFCEE 的发现
		0x01	开启 NFCEE 的发现
		0x02～0xFF	预留

NFCC 收到 NFCEE _ DISCOVER _ CMD 后，向 DH 返回 NFCEE _ DISCOVER _ RSP，如表 10-47 所示。NFCEE _ DISCOVER _ RSP 中携带了状态参数和表示 NFCEE 个数的参数。如果 NFCEE 发现过程失败，那么状态参数为 STATUS _ FAILED，NFCEE 个数设置为 0；否则，状态参数为 STATUS _ OK，并指示连接到 NFCC 上的 NFCEE 个数。

表 10-47 NFCEE _ DISCOVER _ RSP

负 载 域	长 度	值/描述
状态	1 字节	见表 10-56
NFCEE 的个数	1 字节	NFCC 上连接的 NFCEE 个数，取值范围为 0～255

NFCC 为每一个连接到 NFCC 上的 NFCEE 分配一个 NFCEE ID，如表 10-48 所示。

表 10-48　NFCEE ID

值	描　述	值	描　述
0	DH-NFCEE	255	预留
1～254	由 NFCC 动态分配		

如果至少有一个 NFCEE 连接到 NFCC 上，那么对于每一个连接的 NFCEE，NFCC 都会向 DH 发送 NFCEE _ DISCOVER _ NTF 消息，如表 10-49 所示。该通告消息中包含 NFCEE ID、NFCEE 的状态、NFCEE 所支持的协议，以及 NFCEE 的一些额外信息。在 NFCEE 发现过程中，如果 NFCEE 发现过程被开启，任何 NFCEE 的重新连接或移除都会通过 NFCEE _ DISCOVER _ NTF 消息通告给 DH。如果 NFCEE _ DISCOVER _ NTF 中的 NFCEE 状态为 0x02（NFCEE 移出），NFCEE 协议信息数将被设置为 0，并且 DH 到该 NFCEE 的逻辑连接也将被自动关闭，相应的 NFCEE 接口（协议）立即被去激活。如果 NFCEE 发现过程被关闭，任何 NFCEE 状态的变化都不会向 DH 通知。

表 10-49　NFCEE _ DISCOVER _ NTF

负　载　域	长　度	值/描述		
NFCEE ID	1 字节	见表 10-48，不包括 DH-NFCEE		
NFCEE 状态	1 字节	0x00	NFCEE 已连接，已开启	
		0x01	NFCEE 已连接，已关闭	
		0x02	NFCEE 移除	
		0x03～0xFF	预留	
NFCEE 协议的个数	1 字节	所支持 NFCEE 协议的个数		
NFCEE 协议[0…n]	n 字节	见表 10-50		
NFCEE 信息 TLV 的个数	1 字节	TLV 的个数		
NFCEE I 信息 TLV[0…m]	x+2 字节	类型	1 字节	见表 10-51
		长度	1 字节	值的长度
		值	x 字节	类型所对应的值，见表 10-51

NFCEE 协议和 NFCEE 接口是一对一的映射关系，因此定义了一个通用的表来表示 NFCEE 协议和 NFCEE 接口，如表 10-50 所示。具体协议和接口将在 10.7 节介绍。

表 10-50　NFCEE 协议和接口

NFCEE 协议/接口值	定　义
0x00	APDU
0x01	HCI
0x02	类型 3 标签命令集
0x03	透明
0x04～0x7F	预留
0x80～0xFE	私有
0xFF	预留

NFCEE _ DISCOVER _ NTF 中还携带 NFCEE 的一些额外信息，用 TLV 的形式表示，如表 10-51 所示。

表 10-51 TLV

类　型	长　度	值
0x00	n	硬件/注册标识，主要用于 DH 验证 NFCEE 的合法性，由 NFCEE 厂商定义
0x01	n	ATR 字节。当 NFCEE 与 DH 使用 ISO7816 协议通信时，通过 ATR 描述 NFCEE 支持的传输参数，如 T=0 和 T=1 等。另外，可以包括传输速率、版本号、NFCEE 的序列号、NFCEE 的硬件参数等。ATR 可以参照 ISO7816-3
0x02	9～169	T3T 命令集接口补充信息 如果支持 T3T 命令集补充信息，应该包括 PMm、SC、IDm 信息，参见第 4 章的介绍
0x03～0x9F		预留
0xA0～0xFF		私有

10.6.2 NFCEE 的开启和关闭

DH 向 NFCC 发送 NFCEE _ MODE _ SET _ CMD 命令来开启或关闭 NFCEE，如表 10-52 所示。当 NFCEE 被关闭后，NFCC 不会路由任何数据到 NFCEE，并且不会为该 NFCEE 提供与远端 NFC 端点的射频通信。另外，DH 到该 NFCEE 的逻辑连接也被关闭，相应的 NFCEE 接口也被去激活。

表 10-52 NFCEE _ MODE _ SET _ CMD

负　载　域	长　度	值/描述	
NFCEE ID	1 字节	见表 10-48	
NFCEE 模式	1 字节	0x00	关闭 NFCEE
		0x01	开启 NFCEE
		0x02～0xFF	预留

当 NFCC 收到 NFCEE _ MODE _ SET _ CMD 后，向 DH 响应 NFCEE _ MODE _ SET _ RSP 消息，如表 10-53 所示。

表 10-53 NFCEE _ MODE _ SET _ RSP

负　载　域	长　度	值/描述
状态	1 字节	见表 10-56

10.7 NFCEE 接口

10.7.1 NFCEE 接口架构

NFCEE 接口用于 DH 和 NFCEE 通信。与 RF 接口类似，NFCEE 接口也对应 NFCEE 协议，如图 10-26 所示。DH 通过 NFCEE 接口与 NFCEE 进行数据交换。

DH 通过 NFCEE 发现过程来获取 NFCEE 支持的 NFCEE 接口。由于 NFCEE 接口与

NFCEE 协议是一一对应关系，因此 NFCEE _ DISCOVER _ NTF 中的 NFCEE 协议也代表了 NFCEE 支持的 NFCEE 接口。NFCEE 接口与 NFCEE 协议的分类如表 10-50 所示。

NCI 规范中没有定义 NFCEE 接口的激活和去激活命令。从 NFCEE _ DISCOVER _ NTF 获取的 NFCEE ID 和 NFCEE 协议用于逻辑连接的创建。当 DH 通过 CORE _ CONN _ CREATE _ CMD 创建与 NFCEE 的逻辑连接时，就自动激活了 NFCEE 接口。DH 发送 CORE _ CONN _ CLOSE _ CMD 关闭逻辑连接的同时，去激活了 NFCEE 接口。如果 NFCEE 接口激活失败，NFCC 将把 CORE _ CONN _ CREATE _ RSP 消息中的状态参数设置为 NFCEE _ INTERFACE _ ACTIVATION _ FAILED。对于一个 NFCEE 来说，只能同时有一个 NFCEE 接口被激活，也就是说，在 DH 和一个 NFCEE 之间只能有一个逻辑连接。

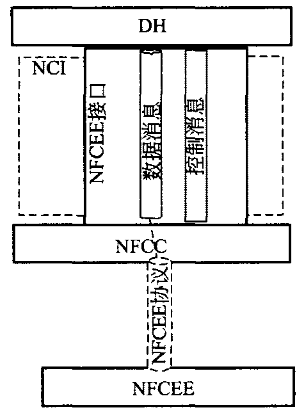


图 10-26 NFCEE 协议与 NFCEE 接口

10.7.2 APDU NFCEE 接口

前面提到过，NFCEE 的具体实现形式可以是各种形式的智能卡，如 SD 卡、SIM 卡等，因此 NFCEE 也支持一些智能卡协议，如 ISO 7816-3。关于 ISO 7816 的介绍请参考第 6 章的 6.5 节。APDU NFCEE 接口是指 NFCEE 支持 ISO 7816-3（交换行业间指令）协议，DH 和 NFCEE 之间通过 APDU 进行数据交换。

DH 通过 APDU NFCEE 接口发送 NCI 数据消息。该 NCI 数据消息的负载中包含 APDU 命令。NFCC 取出 NCI 数据消息中的 APDU 命令，然后将该 APDU 命令发送给 NFCEE，也可以是多个 NCI 数据消息的负载承载了一个 APDU 指令，如图 10-27 所示。

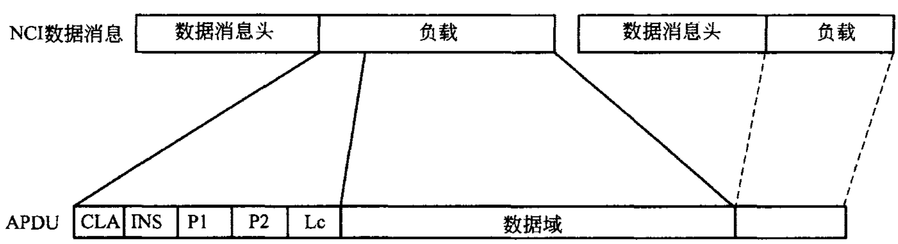


图 10-27 NCI 数据映射进 APDU 命令

当 NFCC 接收到从 NFCEE 发送的 APDU 响应后，把 APDU 响应映射成 NCI 数据消息发送给 DH，如图 10-28 所示。

10.7.3 HCI NFCEE 接口

HCI NFCEE 接口是指 NFCC 与 NFCEE 之间通过 HCI 协议通信。HCI(Host Control Interface) 又称主机控制接口协议，即 ETSI TS 102 622 规范，全称为“UICC-Contactless Front-end(CLF)Interface; Host Controller Interface(HCI)”。该规范定义了 UICC 与 CLF 之间的通信协议。应用在 NCI 规范中，CLF 对应 NFCC，UICC 对应 NFCEE。以手机为例，HCI 就是 SIM 卡与 NFC 芯片之间的通信协议。HCI 定义了主机之间的通信协议，每一个主机都有一个 HOST ID。所有的主机组成一个 HCI 网络。HCI 网络由主机控制器管理，主机

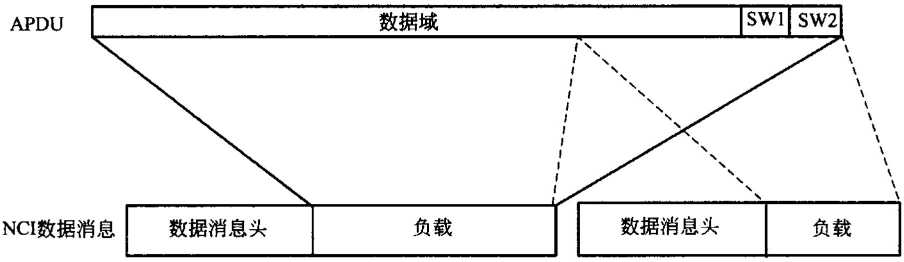


图 10-28　APDU 响应映射进 NCI 数据

之间通过 HCP 数据包通信。

与 APDU 接口类似，DH 通过 HCI NFCEE 接口发送 NCI 数据消息，数据消息的负载中包含 HCP 数据包。NFCC 取出 NCI 数据消息中的 HCP 数据包，发送到 HCI 网络中的 NFCEE。当 NFCC 收到从 NFCEE 发来的 HCP 数据包后，将 HCP 数据包映射进 NCI 数据消息并发送给 DH。

需要注意的是，NCI 1.0 规范不能解决多个 NFCEE 与 NFCC 通过 HCI 相连的场景。也就是说，当存在多个 NFCEE 通过 HCI 网络与 NFCC 相连时，DH 无法寻址到 HCI 网中的 NFCEE。当 NFCEE 与 NFCC 通过 HCI 连接时，无论 HCI 网络中有多少个 NFCEE，NFCC 只会为 HCI 网络分配一个 NFCEE ID。当只有一个 NFCEE 时，这样的寻址没有问题，但是当存在多个 NFCEE 时，DH 无法通过一个 NFCEE ID 来区分不同的 NFCEE。在 HCI 网络中，每一个主机都有一个 Host ID。如何将一个 NFCEE ID 与多个 Host ID 映射，使得 DH 寻址到每个 NFCEE，是目前 NCI 工作组和 ETSI 工作组合作的重点。在 NCI 1.0 设计之初，没有考虑到多个 NFCEE 的情况，但是随着移动支付的发展，多 SE(Security Element) 的方案越来越流行，使得 NCI 工作组不得不尽快解决多 NFCEE 的寻址问题。关于移动支付以及 SE 的概念，将在第 14 章详细介绍。

10.7.4 支持类型 3 标签命令集 NFCEE 接口

NFCEE 可以作为一个类型 3 的标签，DH 可以作为一个类型 3 标签的读卡器对 NFCEE 进行读写操作。比如，DH 向 NFCEE 发送 SENSF _ REQ，NFCEE 向 DH 返回 SENSF _ RES。

DH 通过类型 3 标签命令集 NFCEE 接口向 NFCC 发送 NCI 数据消息。该 NCI 数据消息的负载中包含类型 3 标签的命令。NFCC 收到 NCI 数据消息后，取出类型 3 标签命令，发送给 NFCEE。NFCEE 收到类型 3 标签命令后，向 NFCC 返回类型 3 标签响应；NFCC 将类型 3 标签响应映射进 NCI 数据消息，发送给 DH，如图 10-29 所示。

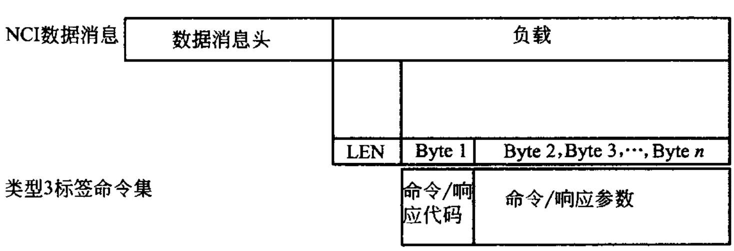


图 10-29　类型 3 标签命令集与 NCI 数据消息

10.7.5　透明 NFCEE 接口

透明 NFCEE 接口是指 DH 与 NFCEE 之间交换数据，NFCC 只转发这些数据，没有任何修改。DH 通过透明 NFCEE 接口向 NFCC 发送 NCI 数据消息，NFCC 收到并取出负载后直接转发给 NFCEE；当 NFCC 收到 NFCEE 发送的数据后，将数据直接封装在 NCI 数据消息中发送给 DH。

10.8　参考附录表

表 10-54　GID 和 OID 的定义

GID	OID	消息名称
NCI Core 0000b	000000b	CORE_RESET_CMD CORE_RESET_RSP CORE_RESET_NTF
	000001b	CORE_INIT_CMD CORE_INIT_RSP
	000010b	CORE_SET_CONFIG_CMD CORE_SET_CONFIG_RSP
	000011b	CORE_GET_CONFIG_CMD CORE_GET_CONFIG_RSP
	000100b	CORE_CONN_CREATE_CMD CORE_CONN_CREATE_RSP
	000101b	CORE_CONN_CLOSE_CMD CORE_CONN_CLOSE_RSP
	000110b	CORE_CONN_CREDITS_NTF
	000111b	CORE_GENERIC_ERROR_NTF
	001000b	CORE_INTERFACE_ERROR_NTF
	001001b～111111b	预留
射频管理 0001b	000000b	RF_DISCOVER_MAP_CMD RF_DISCOVER_MAP_RSP
	000001b	RF_SET_LISTEN_MODE_ROUTING_CMD RF_SET_LISTEN_MODE_ROUTING_RSP
	000010b	RF_GET_LISTEN_MODE_ROUTING_CMD RF_GET_LISTEN_MODE_ROUTING_RSP RF_GET_LISTEN_MODE_ROUTING_NTF
	000011b	RF_DISCOVER_CMD RF_DISCOVER_RSP RF_DISCOVER_NTF
	000100b	RF_DISCOVER_SELECT_CMD RF_DISCOVER_SELECT_RSP
	000101b	RF_INTF_ACTIVATED_NTF
	000110b	RF_DEACTIVATE_CMD RF_DEACTIVATE_RSP RF_DEACTIVATE_NTF

续表

GID	OID	消息名称
	000111b	RF_FIELD_INFO_NTF
	001000b	RF_T3T_POLLING_CMD RF_T3T_POLLING_RSP RF_T3T_POLLING_NTF
	001001b	RF_NFCEE_ACTION_NTF
	001010b	RF_NFCEE_DISCOVERY_REQ_NTF
	001011b	RF_PARAMETER_UPDATE_CMD RF_PARAMETER_UPDATE_RSP
	001100b	RF_LLCP_SYMMETRY_START_CMD RF_LLCP_SYMMETRY_START_RSP
	001101b	RF_LLCP_SYMMETRY_STOP_CMD RF_LLCP_SYMMETRY_STOP_RSP
	001110b	RF_AGGREGATE_ABORT_CMD RF_AGGREGATE_ABORT_RSP
	001111b～111111b	预留
NFCEE 管理 0010b	000000b	NFCEE_DISCOVER_CMD NFCEE_DISCOVER_RSP NFCEE_DISCOVER_NTF
	000001b	NFCEE_MODE_SET_CMD NFCEE_MODE_SET_RSP
	000010b～111111b	预留
私有 1111		

表 10-55 配置参数

参数名称	ID
通用的发现参数	
TOTAL_DURATION	0x00
CON_DEVICES_LIMIT	0x01
CON_DISCOVERY_POLL	0x02
预留	0x03～0x07
轮询模式-NFC-A 发现参数	
PA_BAIL_OUT	0x08
预留	0x09～0x0F
轮询模式-NFC-B 发现参数	
PB_AFI	0x10
PB_BAIL_OUT	0x11
PB_ATTRIB_PARAM1	0x12
PB_SENSB_REQ_PARAM	0x13
预留	0x14～0x17

续表

参数名称	ID
轮询模式-NFC-F 发现参数	
PF_BIT_RATE	0x18
PF_RC_CODE	0x19
预留	0x1A～0x1F
轮询模式-ISO-DEP 发现参数	
PB_H_INFO	0x20
PI_BIT_RATE	0x21
预留	0x22～0x27
轮询模式-NFC-DEP 发现参数	
PN_NFC_DEP_SPEED	0x28
PN_ATR_REQ_GEN_BYTES	0x29
PN_ATR_REQ_CONFIG	0x2A
预留	0x2B～0x2F
侦听模式-NFC-A 发现参数	
LA_BIT_FRAME_SDD	0x30
LA_PLATFORM_CONFIG	0x31
LA_SEL_INFO	0x32
LA_NFCID1	0x33
预留	0x34～0x37
侦听模式-NFC-B 发现参数	
LB_SENSB_INFO	0x38
LB_NFCID0	0x39
LB_APPLICATION_DATA	0x3A
LB_SFGI	0x3B
LB_ADC_FO	0x3C
预留	0x3D～0x3F
侦听模式-NFC-F 发现参数	
LF_T3T_IDENTIFIERS_1	0x40
LF_T3T_IDENTIFIERS_2	0x41
LF_T3T_IDENTIFIERS_3	0x42
LF_T3T_IDENTIFIERS_4	0x43
LF_T3T_IDENTIFIERS_5	0x44
LF_T3T_IDENTIFIERS_6	0x45
LF_T3T_IDENTIFIERS_7	0x46
LF_T3T_IDENTIFIERS_8	0x47
LF_T3T_IDENTIFIERS_9	0x48
LF_T3T_IDENTIFIERS_10	0x49
LF_T3T_IDENTIFIERS_11	0x4A

续表

参数名称	ID
侦听模式-NFC-F 发现参数	
LF_T3T_IDENTIFIERS_12	0x4B
LF_T3T_IDENTIFIERS_13	0x4C
LF_T3T_IDENTIFIERS_14	0x4D
LF_T3T_IDENTIFIERS_15	0x4E
LF_T3T_IDENTIFIERS_16	0x4F
LF_PROTOCOL_TYPE	0x50
LF_T3T_PMM	0x51
LF_T3T_MAX	0x52
LF_T3T_FLAGS	0x53
LF_CON_BITR_F	0x54
LF_ADV_FEAT	0x55
预留	0x56～0x57
侦听模式-ISO-DEP 发现参数	
LI_FWI	0x58
LA_HIST_BY	0x59
LB_H_INFO_RESP	0x5A
LI_BIT_RATE	0x5B
预留	0x5C～0x5F
侦听模式-NFC-DEP 发现参数	
LN_WT	0x60
LN_ATR_RES_GEN_BYTES	0x61
LN_ATR_RES_CONFIG	0x62
预留	0x63～0x7F
其他参数	
RF_FIELD_INFO	0x80
RF_NFCEE_ACTION	0x81
NFCDEP_OP	0x82
LLCP_VERSION	0x83
AGG_INTF_CONFIG	0x84
NFCC_CONFIG_CONTROL	0x85
预留	0x86～0x9F
预留为私有	
预留	0xA0～0xFE
预留用于扩展	
预留	0xFF

表 10-56　状态参数

状态代码	描述
通用状态代码	
0x00	STATUS_OK
0x01	STATUS_REJECTED
0x03	STATUS_FAILED
0x04	STATUS_NOT_INITIALIZED
0x05	STATUS_SYNTAX_ERROR
0x06	STATUS_SEMANTIC_ERROR
0x07～0x08	预留
0x09	STATUS_INVALID_PARAM
0x0A	STATUS_MESSAGE_SIZE_EXCEEDED
0x0B～0x10	预留
0x11	STATUS_OK_1_BIT
0x12	STATUS_OK_2_BIT
0x13	STATUS_OK_3_BIT
0x14	STATUS_OK_4_BIT
0x15	STATUS_OK_5_BIT
0x16	STATUS_OK_6_BIT
0x17	STATUS_OK_7_BIT
0x18～0x9F	预留
RF 发现相关的状态代码	
0xA0	DISCOVERY_ALREADY_STARTED
0xA1	DISCOVERY_TARGET_ACTIVATION_FAILED
0xA2	DISCOVERY_TEAR_DOWN
0xA3～0xAF	预留
RF 接口相关的状态代码	
0x02	RF_FRAME_CORRUPTED
0xB0	RF_TRANSMISSION_ERROR
0xB1	RF_PROTOCOL_ERROR
0xB2	RF_TIMEOUT_ERROR
0xB3	RF_UNEXPECTED_DATA
0xB4～0xBF	预留
NFCEE 接口相关的状态代码	
0xC0	NFCEE_INTERFACE_ACTIVATION_FAILED
0xC1	NFCEE_TRANSMISSION_ERROR
0xC2	NFCEE_PROTOCOL_ERROR
0xC3	NFCEE_TIMEOUT_ERROR
0xC4～0xDF	预留
私有状态代码	
0xE0～0xFF	私有

10.9 小结

NCI 规范是 NFC 协议中内容比较丰富且比较复杂的一个规范。NCI 定义了 DH 和 NFCC 之间的接口，主要目的是标准化 DH 和 NFCC 之间的通信接口，使 DH 能够通过标准化的接口来控制不同厂商的 NFC 芯片。NCI 屏蔽了不同厂商 NFC 芯片实现的细节以及差异化。本章首先对 NCI 的整体架构进行了概述，分别介绍了 NCI 功能模块、NCI 核心以及传输映射层的主要作用；之后，详细描述 NCI 架构中的每一个模块，特别是 RF 发现、RF 接口、RF 协议等重点内容；最后，介绍 NFCEE 的相关内容，包括 NFCEE 的发现过程以及 NFCEE 接口。

Retro Workshop
果粉工作室

第11章 PHDC（个人健康设备通信）规范

11.1 PHDC 概述

在医疗过程中，经常需要将患者当前的生理指标和状态结合历史治疗数据进行综合分析，以便医生做出准确判断和治疗。ISO/IEEE 11073 系列标准定义了医疗健康设备与外部计算机系统之间的通信协议，能够将个人健康设备上的用户数据传输到如医生办公室、诊所或录入监测服务机构等。ISO/IEEE 11073 中定义了两种实体：代理（Agent）和管理者（Manager）。代理的主要作用是收集个人健康数据，并将这些数据发送给关联的管理者。管理者的主要作用就是从一个或多个代理那里获取个人健康数据，如图 11-1 所示。举例来说，管理者可以是手机、健康监视设备、个人电脑等；代理可以是具有传感器的各种医疗设备，如心血管健康检测仪、睡眠监控机、计步器、血压计等。

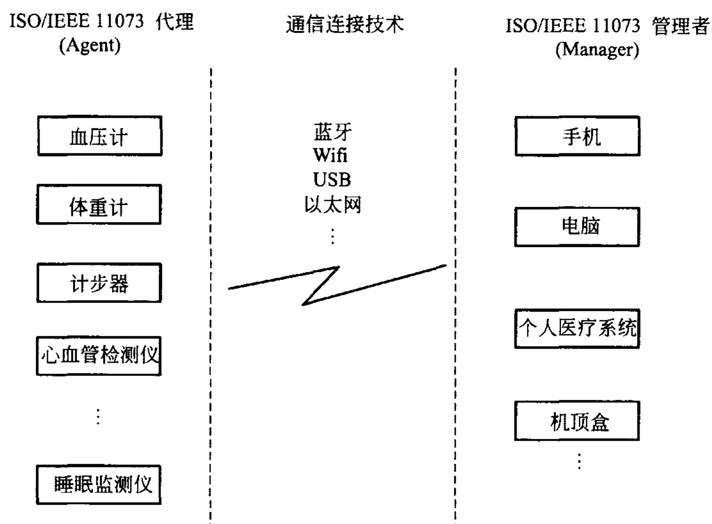


图 11-1 ISO/IEEE 11073 架构

ISO/IEEE 11073 系列标准中定义了各种医疗健康相关参数的数据格式以及代理和管理者之间的通信协议。在 ISO/IEEE 11073 的系列标准中，ISO/IEEE 11073-20601 定义了一

个通用的框架，并对个人健康数据进行抽象，使得数据交换与底层通信技术无关。目前，ISO/IEEE 11073 系列标准已经在具有蓝牙、Zigbee、WiFi 等通信技术的个人医疗设备中应用。NFC 作为一个便捷的非接触通信技术，不仅能够实现设备之间数据快速、安全地传输，还能通过移动互联设备将实时采集的数据与患者历史治疗记录进行综合和智能分析，实现自动化、智能化的实时诊疗服务。例如，把 NFC 芯片内嵌到血糖仪或者创可贴等产品内，监视各种重要的人体生命特征，不仅能了解病人之前的治疗情况，还可查询与病人相关的病例，促进医疗信息的智能管理。

因此，NFC Forum 与 Continua 健康联盟（Continua Health Alliance）紧密合作，发布了个人健康设备通信 PHDC(Personal Health Device Communication）规范[1]。PHDC 规范定义了如何通过 NFC 的点对点通信模式和读写模式传输个人医疗数据。11.2 节和 11.3 节将针对 NFC 设备的这两种工作模式介绍 PHDC 通信流程及消息定义。

11.2 点对点通信模式

11.2.1 PHDC 点对点通信架构

在 NFC 点对点通信模式下，IEEE 11073-20601 的 APDU 指令通过 NFC LLC 数据链路连接进行交换，如图 11-2 所示。PHDC 代理和 PHDC 管理者是 PHDC 规范定义的功能组件，分别对应 IEEE 11073 中的代理和管理者。PHDC 代理主要负责将 IEEE 代理的 PDU 封装成 PHDC PDU 发送给 PHDC 管理者，并接收 PHDC 管理者发来的 PHDC PDU，解析出 IEEE APDU 后通知给 IEEE 代理。PHDC 管理者负责将 IEEE 管理者的 APDU 封装成 PHDC PDU 发送给 PHDC 代理，并接收 PHDC 代理发来的 PHDC PDU，解析出 IEEE APDU 后通知给 IEEE 管理者。从该架构图可以看出，PHDC 实际上是定义了一个适配层，能够在 NFC 数据链路完成 IEEE APDU 的收发。另外，PHDC 代理和管理者负责对 IEEE APDU 指令分段和重组。

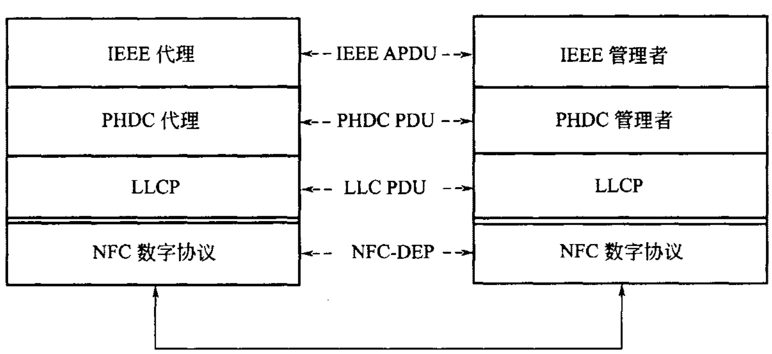


图 11-2 PHDC 点对点通信模式架构

在 NFC 点对点模式下，一个或多个 PHDC 代理可以与一个 PHDC 管理者通信，如图 11-3 所示。每一个代理和管理者之间都有一个 LLC 逻辑数据链路连接。

PHDC 在 LLCP 中注册的服务名称为“urn：nfc：sn：phdc”和“urn：nfc：sn：

[1] NFC Forum. Personal Health Device Communication Technical Specification, PHDC 1.0.

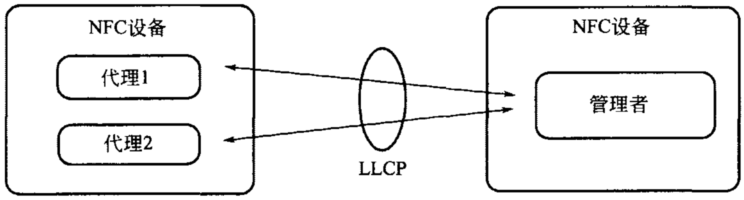


图 11-3　PHDC 点对点通信拓扑结构

phds”。其中，“urn：nfc：sn：phdc”用于非加密的 PHDC 通信，“urn：nfc：sn：phds”用于加密的 PHDC 通信。

11.2.2　数据格式定义

PHDC 代理和 PHDC 管理者以 PHDC PDU 的方式交换数据。PHDC PDU 的数据格式如图 11-4 所示。PHDC PDU 头中包含一个 16bit 的值，用于表示信息域的长度。信息域中包含一个 IEEE APDU。

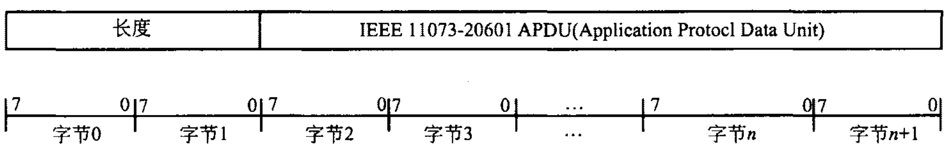


图 11-4　PHDC PDU 结构

11.2.3　通信流程

当 LLC 数据链路激活后，LLC 将通知 PHDC 管理者 LLC 链路已经激活；接下来，PHDC 管理者执行自己的激活过程。首先，PHDC 管理者会分配一个本地的 SAP，并将该 SAP 与 PHDC 的服务名称相关联；之后，等待对该 SAP 的数据链路连接请求。对 PHDC 代理来说，当 LLC 链路激活后，PHDC 将执行自己的激活流程。PHDC 代理会试图建立与 PHDC 管理者的 LLC 数据链路连接，或者是通过 LLCP 的 SDP 协议来发现 PHDC 管理者的 SAP。

在进入正常的数据通信过程中，PHDC 管理者向本地 IEEE 管理者发送传输连接指示，表明数据连接已经建立好，IEEE 管理者可以发送 APDU 了；PHDC 代理也会向本地 IEEE 代理发送传输连接指示，表明数据连接建立成功。在通信过程中，如果 PHDC PDU 的长度超出了 LLC I PDU 所能承载数据的最大长度，PHDC PDU 将被分段传输。图 11-5 描述了分段的过程。在接收的 PHDC 侧，负责将分段的 PHDC PDU 重组。

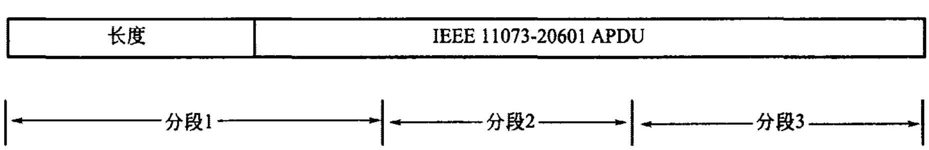


图 11-5　PHDC PDU 分段

图 11-6 给出了一个简单的 PHDC PDU 交换的例子。注意，本例中省略了将 PHDC PDU 映射进 LLC I PDU 的过程。实际上，PHDC PDU 就承载在 I PDU 的信息域中。图中给出了从连接建立、数据传输到连接释放的全过程。

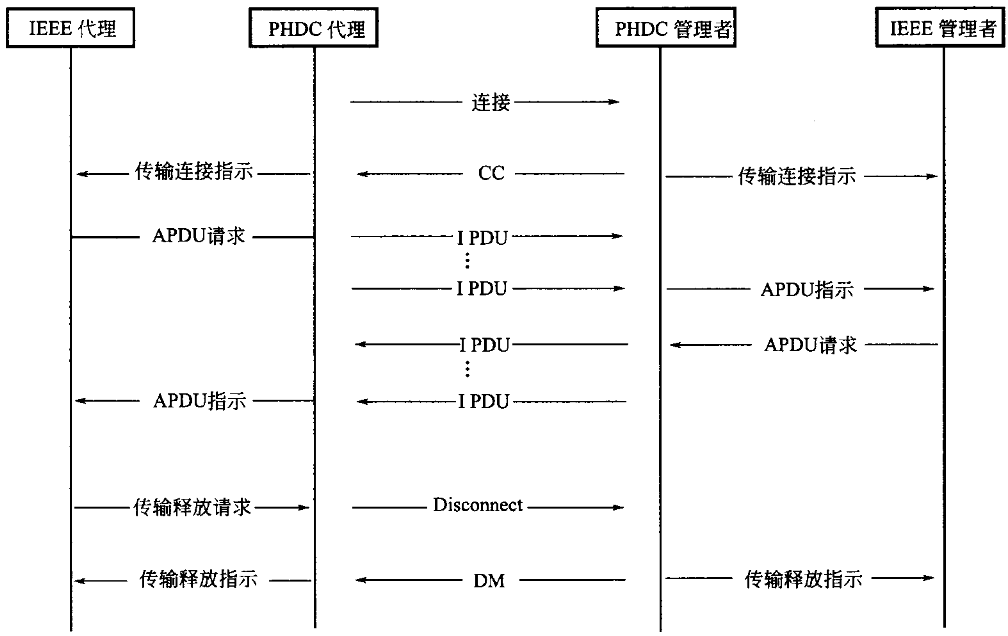


图 11-6 PHDC 点对点通信示例

11.3 读写模式

11.3.1 PHDC 读写模式通信架构

对于 PHDC 规范中的 NFC 读写模式，IEEE APDU 封装在 NDEF Record 中，NFC 读写器从标签中读取或向标签中写入 NDEF Record，如图 11-7 所示。

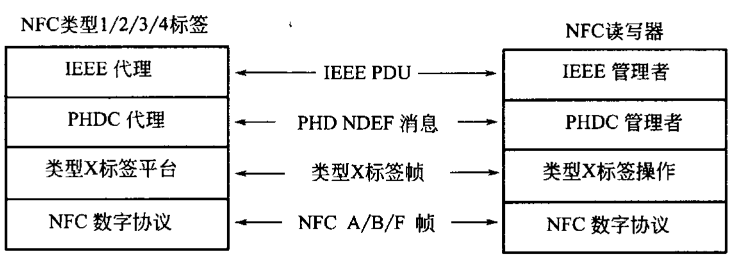


图 11-7 PHDC 读写模式架构

在读写模式下，PHDC 管理者位于 NFC 设备中，PHDC 代理位于 NFC 标签中。PHDC 管理者与 PHDC 代理之间通过标签操作规范中定义的命令和响应通信，如图 11-8 所示。

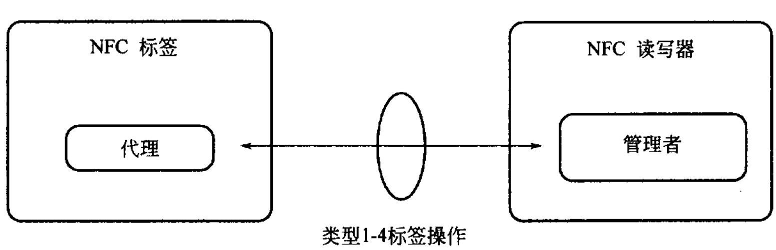


图 11-8 PHDC 读写模式拓扑结构

11.3.2 数据格式定义

PHD(Personal Health Device) 消息用于 PHDC 代理或管理者传输 IEEE 11073-20601 APDU。PHD 消息格式如图 11-9 所示，该消息以 PHD Record 起始，后面跟着一个或多个 NDEF Record。

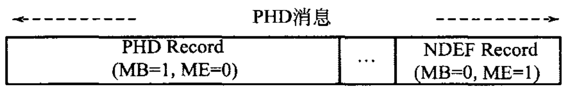


图 11-9　PHD 消息格式

PHD Record 中封装了 IEEE 11073-20601 的 APDU，如图 11-10 所示。PHD Record 的类型为 NFC Forum 已知类型全局类型 “PHD”。第一个字节的 b7 为 LC(Link Connected)，设置为 1，表示要发送的数据紧随在先前收到的数据之后。MC(Message Counter) 表示数据传输过程中数据消息的个数。PHDC 代理向 PHDC 管理者发送的第一个 PHD 消息中将 MC 设置为 0，之后的消息无论是代理发给管理者的还是管理者发给代理的，每一个消息中的 MC 都加 1 并进行模 16 运算。MC 的最低位表示消息的方向。如果 MC 模 2 等于 0，数据是从 PHDC 代理到 PHDC 管理者的；如果 MC 模 2 等于 1，数据是从 PHDC 管理者到 PHDC 代理的。

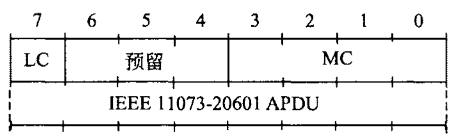


图 11-10　PHD Record

11.3.3 通信流程

PHDC 读写模式的通信流程如图 11-11 所示，PHDC 管理者和 PHDC 代理都可以通过读写命令同时接入 NDEF 数据域，读取 PHD 消息。NDEF 数据域（如 NDEF 标签）与运行 PHDC 代理的主机通过有线连接，当 NDEF 数据域被 PHDC 管理者进行读写操作时，PHDC 代理将收到通知消息。PHDC 代理通过该通知消息判断 PHDC 管理者是否完成了从 NDEF 数据域中读取 PHD 或写入 PHD 消息的操作。尽管管理者和代理可以同时对 NDEF 数据域进行读写，但是每一个读写命令都是原子操作，即在任何时间段内，只能有一个命令在执行，而且该命令的执行不会被其他命令的到来而打断。

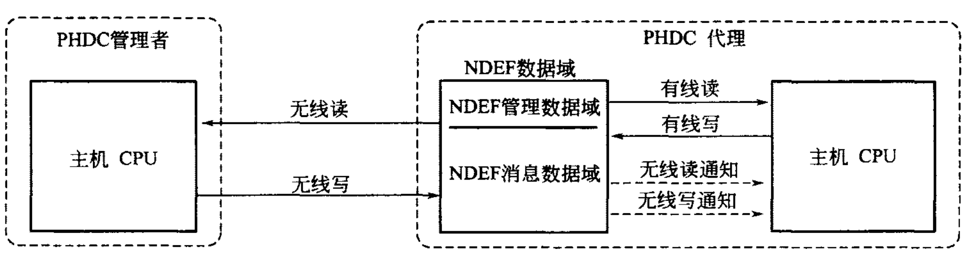


图 11-11　PHDC 读写模式流程

11.4 小结

本章介绍了 NFC 在个人医疗领域的应用场景，并详细介绍了 NFC Forum 制定的 PHDC 规范。该规范定义了如何通过 NFC 的点对点通信模式和读写模式传输个人医疗数据。针对 NFC 设备的这两种工作模式，分别介绍了 PHDC 的通信架构、消息格式定义以及通信流程。

第12章 Connection Handover规范

12.1 Connection Handover 概述

Connection Handover 规范❶，又称为连接切换规范，由 NFC Forum RAF 工作组制定。为简便起见，本章中称其为 CH 规范。CH 规范定义了通过 NFC 技术在两个 NFC 设备间协商和激活其他无线通信技术（如 WiFi、蓝牙）的工作流程以及相应的消息格式。

与 WiFi、蓝牙等无线通信技术相比，NFC 通信技术的连接建立速度非常快，只需要两个 NFC 设备靠近，就能迅速地发现对方并快速地进行数据传输。但是 NFC 不适合传输大量数据的场景。NFC 传输速率不高，需要两个设备靠近后才能传输。如果有大量数据要传输，需要两个设备长时间靠近，用户体验不好。另外，目前很多的应用都是基于 WiFi、蓝牙等通信技术开发的，修改这些应用来支持 NFC 不是一件容易的事情。但是，NFC 通信技术有自己的优势：建立连接很快，而且适合传输少量数据，特别适合两个 NFC 设备间进行其他通信技术的协商和选择。

举例来说，目前大部分手机都支持 WiFi 和蓝牙，如果两个手机之间要传输文件，如何选择通信技术，以及在选择某个通信技术之后如何快速建立通信连接，都可以通过 CH 规范定义的机制来解决。

以手机接入 AP（Access Point，俗称为无线路由器）为例，一般情况下，手机需要获知 AP 的网络名称（SSID）以及密码之后才能接入网络。手机首先要扫描获取 AP 名称，用户需要输入密码后才能接入 AP 网络。这个过程比较繁琐。如果 AP 和手机都支持 NFC 功能，那么通过 CH 规范定义的流程，用户只需要拿着手机触碰一下 AP，AP 所有的网络配置信息，如 AP 的网络名称、加密方式、密码等信息会被传递到手机中，手机就可以自动连接到 AP 网络。

CH 规范也可以用在辅助蓝牙设备的配对中。两个蓝牙设备要建立连接，首先要发现对方并配对，配对成功后才能传输文件。众所周知，蓝牙的配对过程比较慢，如果通过 NFC 技术来辅助蓝牙配对，将大大缩短蓝牙配对时间。通过两个 NFC 设备的触碰，完成蓝牙配置信息的交换，加速了蓝牙配对速度。这个方案已经被广泛应用到各种产品中，正如第 1 章中提到的，如诺基亚具有 NFC 功能的蓝牙耳机与诺基亚手机之间通过触碰就完成了手机与

---

❶ NFC Forum. Connection Handover Technical Specification, Connection Handover 1.2.

蓝牙耳机之间的配对，还有三星的 S Beam 等。

为了更好地支持 NFC 技术来辅助蓝牙配对，NFC Forum 与 Bluetooth SIG 共同制定了 Bluetooth Secure Simple Pairing Using NFC 规范。该规范定义了通过 NFC 来辅助蓝牙完成安全的、简单的、配对的实施方案，让产品开发者更快、更容易地实现 NFC 辅助蓝牙配对的功能。本章对该规范不再详述，感兴趣的读者可以在 NFC Forum 官方网站上获取相关信息。

12.2 连接切换流程

CH 规范中定义了两个设备角色，即切换请求方（Handover Requester）和切换选择方（Handover Selector）。切换请求方通过 NFC 通信链路向切换选择方发送切换请求（Handover Request）消息来进行其他通信技术的协商和激活。当切换选择方收到切换请求消息后，回复切换选择（Handover Select）消息。其中包含切换请求方支持的通信技术及相关信息。切换选择消息包含切换选择方支持的通信技术及其相关信息。通过切换请求消息和切换选择消息的交互，两个 NFC 设备获取了彼此支持的通信技术以及相应的配置信息，这样就可以选择某个通信技术建立连接。

通信技术的协商与激活分为两种模式，一种是协商切换，由两个 NFC 设备间通过点对点通信完成。另一种是静态切换，一个 NFC 设备工作在读写模式下，而另一个设备配备有 NFC 标签，标签中存放该设备所支持的通信技术及其相关信息。NFC 设备读取标签中的信息，从而获知该设备所支持的通信方式。下面将分别介绍协商切换和静态切换两种模式。

为了与 CH 规范中的术语保持一致，本章中将其他通信技术（如 WiFi、蓝牙等）统称为可选载波（Alternative Carriers）。

12.2.1 协商切换

协商切换的基本流程如图 12-1 所示。

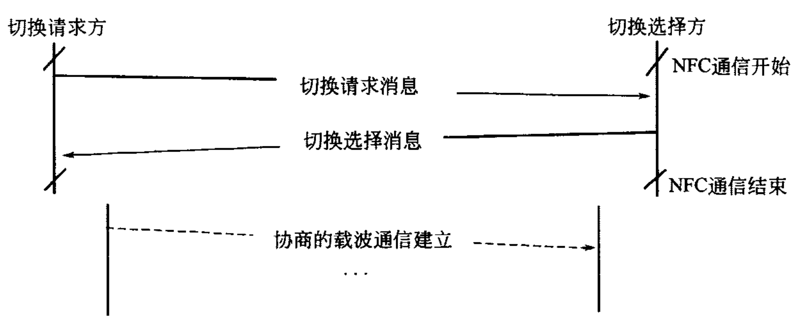


图 12-1 协商切换流程

① 切换请求方向切换选择方发送切换请求消息。该消息中包含切换请求方支持的可选载波的数量、类型等，也包含这些载波的相关配置信息。以 WiFi 为例，这些配置信息包含 AP 的网络名称、密码等。载波的配置信息在切换请求消息中是可选的。

② 切换选择方收到切换请求消息后，比较自己所支持的通信载波技术，并判断所支持的载波技术目前的状态，如关闭还是打开，是否有足够的电量来打开该通信载波技术。切换选择方向切换请求方返回切换选择消息。该消息中包含与切换请求消息中对应的载波技术。如果切换选择方支持的载波技术与切换请求方支持的载波技术无法匹配，或因为某种原因无

法打开所对应的载波技术，切换选择方将返回一个不包含任何可选载波技术的切换选择消息。举例来说，如果切换请求方支持 WiFi，但是切换选择方只支持蓝牙，二者将无法建立数据连接，连接选择方将返回一个空的切换选择消息。

如果切换选择方返回了包含可选载波技术列表的切换选择消息，表示切换协议成功完成，接下来通信的建立取决于两个设备所选择的通信技术，并不在 CH 规范的范围内。

图 12-2 描述了切换请求方如何与切换选择方进行可选载波的协商，并最终选择了一个匹配的可选载波。在该示例中，切换请求方在切换请求消息中宣告自己支持的可选载波为 WiFi 和蓝牙。当切换选择方收到该消息后，将接受蓝牙作为接下来数据传输的可选载波，因此在切换选择消息中携带的可选载波为蓝牙。这里尽管切换选择方也支持 WiFi，但是它可以根据自己的设置来选择通过蓝牙通信。

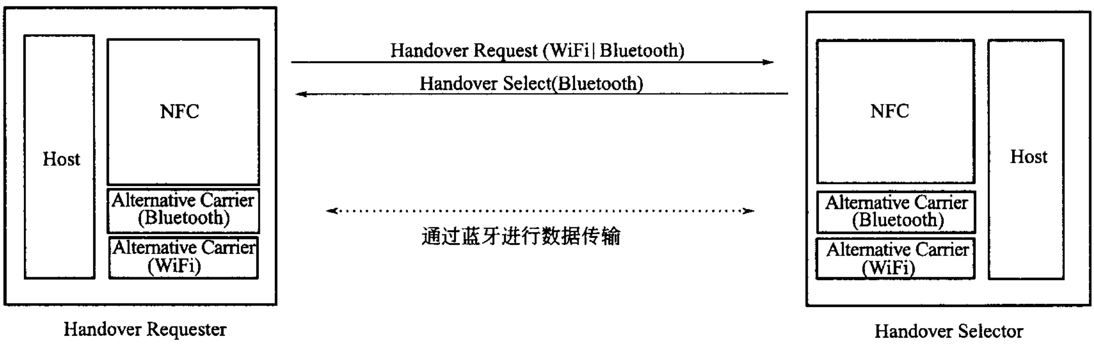


图 12-2 协商切换流程示例-1

图 12-3 描述了切换选择方支持多个可选载波，并在切换选择消息中携带多个可选载波的情况。切换请求方收到该消息后，自由选择其中任何一个载波技术用于接下来的数据通信，甚至可以选择同时建立不同的载波技术连接，如同时建立 WiFi 和蓝牙连接。但是一般情况下，切换选择消息中的可选载波列表实际上隐含着优先级的设定。位于载波列表中的第一个可选载波是应当优先使用的。当然，这不是绝对的，图 12-3 就描述了切换请求方没有按照隐含的优先级执行。图中，切换选择方将 WiFi 放在载波列表的第一位，隐含地表示 WiFi 是第一选择，但是切换请求方最终使用蓝牙完成接下来的数据通信。

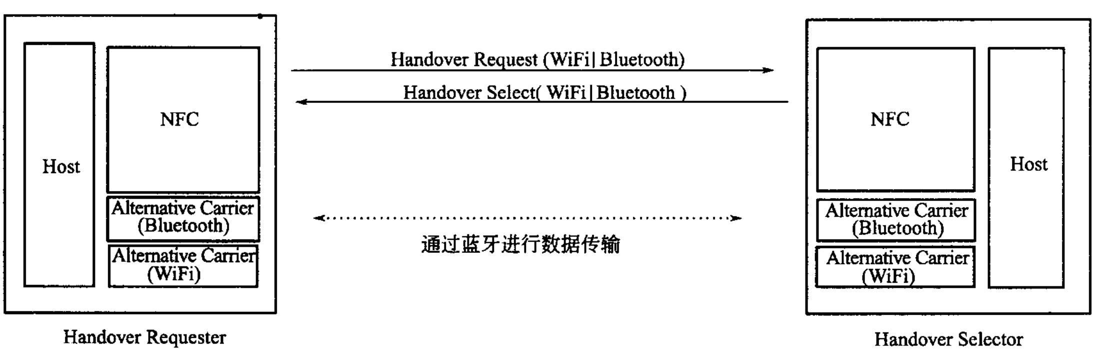


图 12-3 协商切换流程示例 2

在图 12-4 中，切换选择方在切换选择消息中携带可选载波 WiFi 和蓝牙，并将蓝牙放在载波列表中的第一个，认为蓝牙是第一优先级的。但是切换请求方在尝试通过蓝牙建立连接

的过程中失败了（如因为两个设备移动，超出了蓝牙的工作距离），切换请求方决定采用第二选择，即通过 WiFi 建立数据连接。



图 12-4　协商切换流程示例-3

由于电量的一些约束条件，切换选择方可能不会激活所有的可选载波（如蓝牙或 WiFi 处于关闭状态）。在这种情况下，切换选择方可以将可选载波的电量状态标识符设置为未激活（inactive）。切换请求方将接着发送切换请求消息，消息中的载波列表只包含一个可选载波。切换选择方返回切换选择消息进行确认，将可选载波的电量状态标识符设置为激活（active）。如图 12-5 所示，切换选择方在切换选择消息中将蓝牙和 WiFi 的电量状态标识符分别设置为未激活。切换请求方收到该消息后，重新发送切换请求消息，请求消息中只包含一种可选载波技术。



图 12-5　协商切换流程示例-4

在 NFC Forum 的协议栈架构中，连接切换协议基于 LLCP 面向连接的服务，切换请求消息和切换选择消息通过 LLCP 中定义的 I PDU 传输。切换选择方的应用在 LLCP 上注册的服务名称为 urn：nfc：sn：handover，这样，远端的切换请求方就可以通过 SDP 协议来获取到连接切换服务的 SAP。

两个 NFC 设备在 NFC 链路建立后，可能同时发送切换请求消息，即 NFC 设备在发送切换请求消息后，紧接着收到一个切换请求消息，造成切换请求冲突。为了解决这个问题，切换请求消息中会包含一个随机数，用于判定由谁来作为切换请求方。NFC 设备比较自己发出去的切换请求消息中的随机数与接收到的切换请求消息中的随机数大小，判断自己是否作为切换请求方；对于另一个 NFC 设备来说，也通过相同的办法来判断。如果两个随机数相等，两个 NFC 设备将重新发送切换请求消息，其中包含重新产生的随机数。有关比较的细节，将在 12.3 节介绍。

12.2.2 静态切换

静态切换的场景一般适合于切换选择方不具备 NFC 功能，但是可以配有一个 NFC 标签，如图 12-6 所示。NFC 标签提供了用于读写的存储空间，但是 NFC 标签可能没有与主处理器连接的电路，因此 NFC 标签无法接收和处理切换请求消息，并且动态地构成相应的切换选择消息。在这种情况下，标签中会存放一个切换选择消息的数据结构，这样切换请求方作为读写器从标签中直接读取切换选择消息。静态切换存在的主要问题在于标签中的内容是静态的，切换选择方的主处理器无法将载波的相关动态信息，如电量状态标识符等，与标签中的切换选择消息同步。

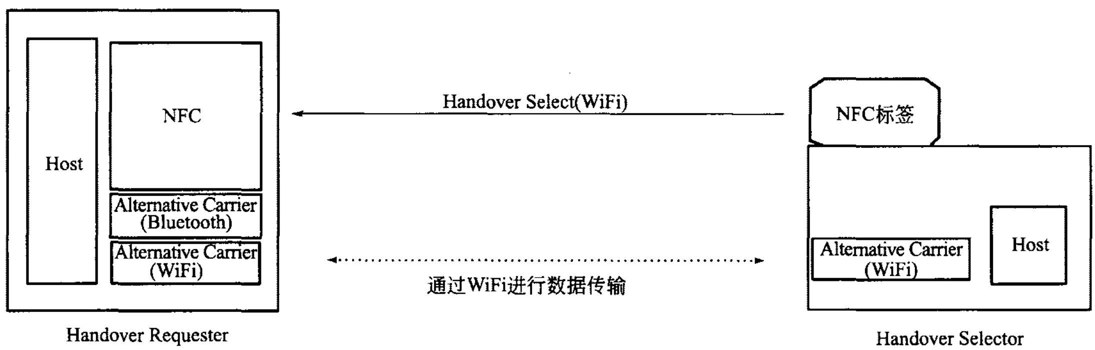


图 12-6 静态切换流程

需要注意的是，在某些应用场景的技术方案中，NFC 标签与主处理器存在连接电路，或者标签放置在一个 NFC 读写器的附近，标签可以作为两个 NFC 读写器之间的通信介质。

12.3 连接切换消息

12.3.1 消息格式

切换消息的结构如图 12-7 所示。切换消息由切换请求 Record，或者是切换选择 Record 组成，后面跟随着任意数量的 NDEF Record。其中，切换请求 Record 的类型为“Hr”，切换选择 Record 的类型为“Hs”，它们均为 NFC Forum 已知类型（TNF＝0x01）的全局类型。

切换请求 Record 或者切换选择 Record 中包含了一组可选载波 Record（Alternative Carrier Record）。可选载波 Record 的类型为 NFC Forum 已知类型的本地类型“ac”。实际上，可选载波 Record 没有包含可选载波的具体内容信息，而是提供了指向包含具体内容的 NDEF Record 的引用。简单来说，切换请求 Record 或者切换选择 Record 后面跟随的 NDEF Record 中包含可选载波的具体内容；而可选载波 Record 中包含 URI 引用，这些引用指向相应的 NDEF Record。一个可选载波 Record 对应一个可选载波，即一个可选载波 Record 代表一种通信技术。图 12-7 中具体的内容域将在 12.3.2 节及 12.3.3 节介绍。

由于可选载波 Record 与其他 NDEF Record 之间为引用关系，因此切换请求消息或切换选择消息的发送方负责创建 Record 中的 ID 域，保证 ID 域的唯一性。

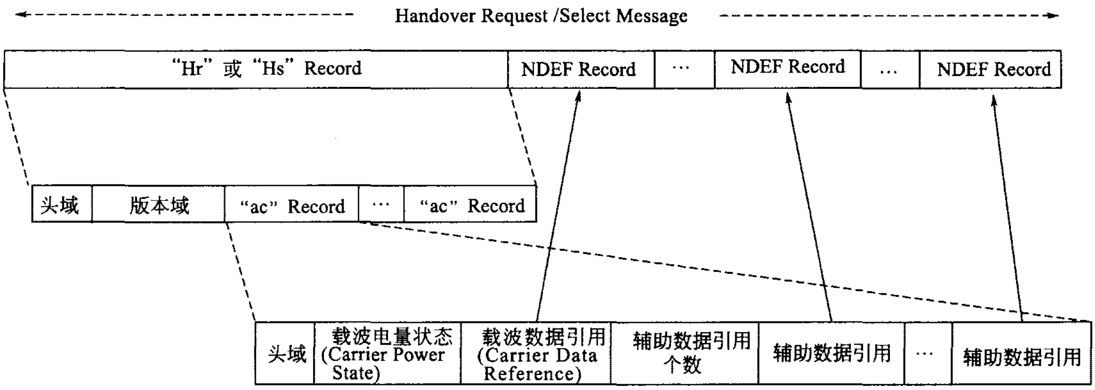


图 12-7 切换消息构成

12.3.2 切换请求消息

切换请求消息的格式如图 12-8 所示。切换请求消息中包含了切换请求方支持的可选载波信息。切换请求消息必须以切换请求 Record 作为消息的开始（MB=1），在切换请求 Record 之后跟随一些 NDEF Record，最后一个 NDEF Record 的 ME 被设置为 1。需要注意的是，切换请求 Record 中的 ME 不能置为 1。换句话说，切换请求消息中必须要包含至少一个 NDEF Record，因为切换请求方只有自己支持可选载波，才会发出切换请求消息；如果不支持任何可选载波，将不发送切换请求消息。

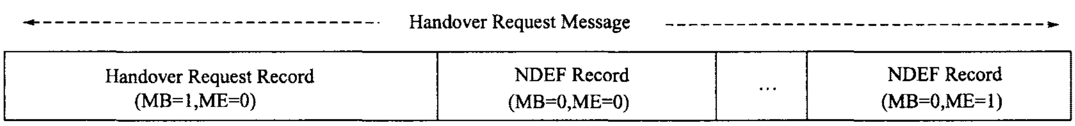


图 12-8 切换请求消息结构

切换请求 Record 的类型为"Hr"，格式如图 12-9 所示，其中包含所支持 CH 规范的版本号、可选载波的列表、用于解决切换请求冲突的 Record。需要注意的是，图 12-9 只是描述了切换请求 Record 的负载域，关于 Record 的头域，如 MB、ME、TNF、负载的长度域、类型的长度域等内容，请参考第 7 章中的描述。本章在描述其他 Record 时，均省略头域。

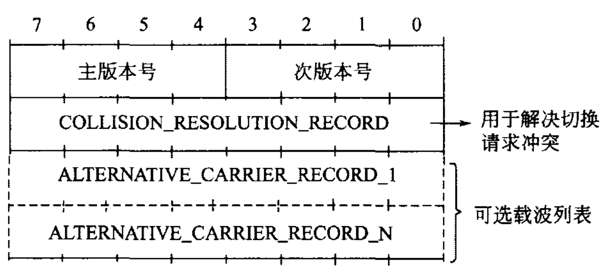


图 12-9 切换请求 Record 结构

本章参考的 CH 规范版本为 1.2，因此主版本号设置为 0x1，次版本号设置为 0x2。

COLLISION _ RESOLUTION _ RECORD 称为冲突解决 Record，类型为 NFC Forum 已知类型的本地类型"cr"，负载域为一个 16bit 随机数，如前面提到的，该 Record 用于解决切换请求冲突，其格式如图 12-10 所示。切换请求冲突解决的步骤如下所述：

① 比较自己发出去的切换请求消息中的随机数与收到的切换请求消息中的随机数。

② 如果两个随机数在数值上相等，两个 NFC 设备将重新产生随机数，并再次发送切换请求消息。

③ 如果两个随机数不相等，首先比较两个随机数的最低位。如果最低位相同，产生较大随机数的 NFC 设备将作为切换选择方，产生较小随机数的 NFC 设备作为切换请求方，并忽略先前收到的切换请求消息。如果两个随机数的最低位不一致，如一个为 0，另一个 1，那么产生较小随机数的设备作为切换选择设备，产生较大随机数的设备为切换请求设备，并忽略先前收到的切换请求消息。

通过最低位比较，能够有效防止某些设备“随机”产生较大值或较小值，甚至是默认的最大值或最小值，从而固定充当切换请求方或切换选择方的问题。



图 12-10 冲突解决 Record 负载域

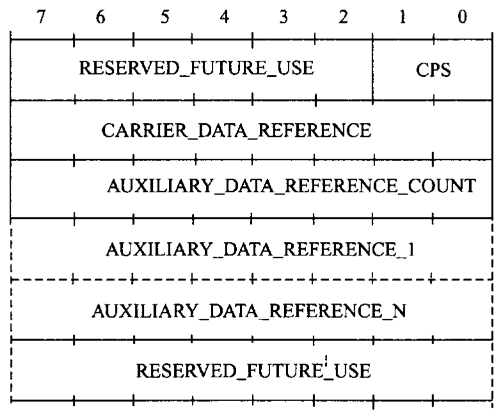


图 12-11 可选载波 Record 负载域

ALTERNATIVE _ CARRIER _ RECORD 称为可选载波 Record，其类型为 NFC Forum 已知类型的本地类型“ac”。该 Record 包含在切换请求 Record 或切换选择 Record 中，用于描述一个可选载波，其格式如图 12-11 所示。

其中，CPS（Carrier Power State）称为载波电量状态，用于指示该载波的电量状态，取值如表 12-1 所示。

表 12-1 CPS

值	描述
0x00	Inactive：表示该载波目前处于关闭状态
0x01	Active：表示该载波已经开启
0x02	Activating：表示该载波处于正在打开的状态，但未开启
0x03	Unknown：未知状态

CARRIER _ DATA _ REFERENCE 是一个指针，如图 12-12 所示，指向一个 NDEF Record。该 Record 唯一地标识一个载波技术。CARRIER _ DATA _ REFERENCE _ LENGTH 表示载波数据引用字符 CARRIER _ DATA _ REFERENCE _ CHAR 的长度，CARRIER _ DATA _ REFERENCE _ CHAR 用于存放 URI 的指针，指向唯一标识载波技术的 NDEF Record。这里所说的唯一标识一个载波技术，是指该 NDEF Record 中包含该载波的类型，可以是 WiFi、蓝牙等。

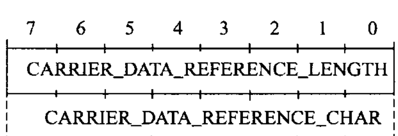


图 12-12 CARRIER _ DATA _ REFERENCE 结构

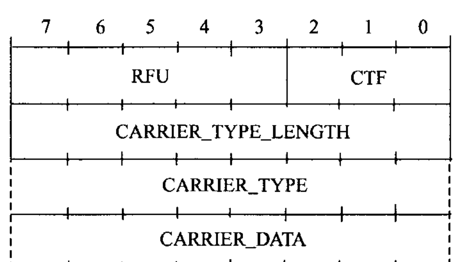


图 12-13 切换载波 Record 结构

如果切换请求消息中没有包含载波的相关配置数据，可选载波 Record 的 CARRIER _ DATA _ REFERENCE 指向的 NDEF Record 称为切换载波 Record（Handover Carrier record），载波的类型为切换载波 Record 负载域中的 CARRIER _ TYPE。切换载波 Record 如图 12-13 所示，其中 CTF（Carrier Type Format）称为载波类型格式，与 TNF 的编码取值一样。

如果切换请求消息中包含载波的相关配置信息，可选载波 Record 的 CARRIER _ DATA _ REFERENCE 应当指向包含载波配置信息的 NDEF Record，载波类型就是该 NDEF Record 的负载类型。由于该 NDEF Record 的负载域中包含该载波的相关配置信息，所以也称该 NDEF Record 为载波配置 Record（Carrier Configuration record），但这只是一个概念上的名字，在 CH 规范中没有专门定义这种类型的 Record。

总的来看，切换载波 Record 只能用在切换请求消息中，而载波配置 Record 可以用于切换请求消息，也可以用于切换选择消息。在切换请求消息中，对于一个载波来说，只能包含切换载波 Record 或载波配置 Record 其中一个。在切换请求消息中包含切换载波 Record 的例子很常见。举例来说，在一般情况下，切换请求方是数据通信的发起方，若手机要接入 AP 的 WiFi 网络，那么手机使用 NFC 通信向 AP 发送切换请求消息。该切换请求消息中只需要告知 AP 手机支持 WiFi 功能即可，无需传输关于手机 WiFi 配置的信息；但是对于手机来说，需要获取 AP 的详尽配置信息才可以接入网络。

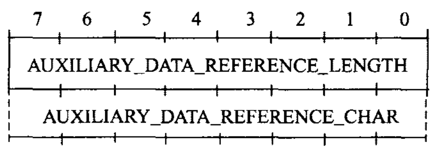


图 12-14 辅助数据应用域结构

可选载波 Record 中的 AUXILIARY _ DATA _ REFERENCE _ COUNT 用于指示辅助数据引用域的个数。AUXILIARY _ DATA _ REFERENCE 是一个指针，指向一个 NDEF Record。该 NDEF Record 提供了可选载波的一些额外信息。辅助数据应用域是可选的，其格式如图 12-14 所示。

12.3.3 切换选择消息

切换选择消息的格式如图 12-15 所示。切换选择方使用切换选择消息来确认收到的切换请求消息。切换选择消息以切换选择 Record 开始，后面跟随 0 个或多个 NDEF Record。如果切换选择 Record 之后没有 NDEF Record，MB 和 ME 均设置为 1。在这种情况下，表示切换选择方不支持切换请求消息中的任何载波。

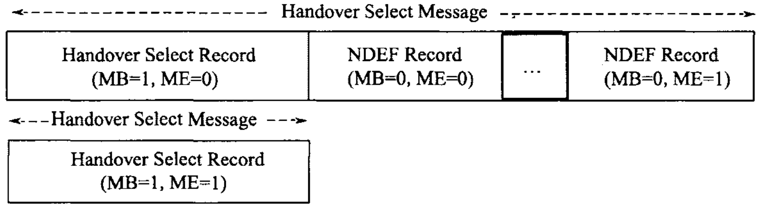


图 12-15 切换选择消息

切换选择 Record 的类型为“Hs”，格式如图 12-16 所示，包含版本号信息、可选载波列表以及错误 Record。其中，版本号信息、可选载波列表中的可选载波 Record 与切换请求

Record 中的含义一致，这里不再赘述。

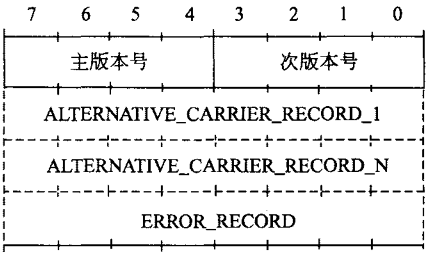


图 12-16 切换选择 Record

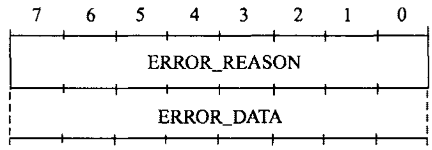


图 12-17 错误 Record

错误 Record（Error Record）的类型为 NFC Forum 已知类型的本地类型“err”，指示了切换选择方不能够处理切换请求消息。不能处理可以是完全不能处理，也可以是部分不能处理。切换选择消息中只能有一个错误 Record，且必须在可选载波 Record 的后面。错误 Record 的负载域如图 12-17 所示。

其中，ERROR _ REASON 为错误原因代码，如表 12-2 所示。

表 12-2 ERROR _ REASON

值	描述
0x00	预留
0x01	由于暂时的内存限制无法处理切换请求消息，在一段时间间隔后重新发送切换请求消息或许可以成功处理
0x02	由于内存限制，无法处理切换请求消息，重新发送切换请求会遇到相同的错误
0x03	由于某个载波的限制，无法处理切换请求消息，在一段时间后重新发送切换请求消息可能也不会处理成功
0x04 - 0xFF	预留

ERROR _ DATA 提供了与错误相关的额外信息，如表 12-3 所示。

表 12-3 ERROR _ DATA

Error Reason	Error_Data 内容
0x01	8bit 无符号整数，用于表示切换选择方发送错误指示到重新接收切换请求消息的最小间隔，用毫秒(ms)表示
0x02	32bit 无符号整数，表示最大能够接收切换请求消息的最大字节数
0x03	8bit 无符号整数，用于表示切换选择方发送错误指示到重新接收切换请求消息的最小间隔，用毫秒(ms)表示

12.4 小结

本章介绍了连接切换规范的主要内容。首先介绍 CH 规范的概要内容，包括制定 CH 规范的目的以及 CH 规范的应用场景；然后，分别介绍连接切换设备的两个角色，连接切换的流程，协商切换和静态切换两种切换模式；最后，详细说明切换请求消息和切换选择消息。

第13章 NFC无线充电技术

13.1 无线充电技术概述

13.1.1 无线充电发展历史

对自由的追求是人类亘古不变的天性，随着科技不断发展，无线通信技术让人们摆脱了位置和距离的束缚，享受随时随地通信的便利性。而在电能传输方面，人们同样释放出对于摆脱线缆的渴望，各种移动电源解决方案层出不穷。科技的发展速度是惊人的，无线充电技术悄然进入人们的视野，特别在消费类电子产品中，具有无线充电的手机已经上市；在汽车领域，电动汽车无线充电解决方案基本成型，虽然还不成熟，但是显示出巨大的市场潜力。

追溯无线充电的历史，要先从无线供电说起。在 19 世纪，对于电力传输有两种思路，一种是以爱迪生为代表的有线派，即架设线缆用于电力的远距离传输，这种方案成熟可靠，缺点是工程量巨大，并且成本高昂；还有一种是以尼古特·特斯拉（如图 13-1 所示）为代表的无线派。早在 1890 年，尼古特·特斯拉就开始构想无线供电的方法❶：把地球作为内导体，距离地面约 60 km 的电离层作为外导体，在地球与电离层之间建立起大约 8Hz 的低频共振，再利用环绕地球的表面电磁波来远距离传输电力，他想象电能可以像广播一样传遍全球。1900 年，特斯拉得到了 150000 美金的研究经费，并开始计划建造华登克里夫塔。这个塔一度被当时的报纸称为“特斯拉的百万大建筑”，如图 13-2 所示。特斯拉希望用这个塔进行跨大西洋的无线电广播和无线电能传输实验。特斯拉最终建成了一座高 187 英尺的铁塔，其顶部有一个直径为 68 英尺的半球形圆顶。

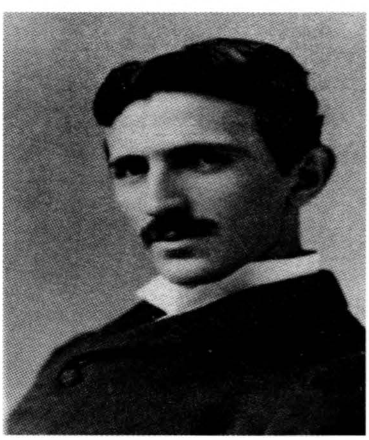


图 13-1　尼古拉·特斯拉

但是 1908 年通古斯大爆炸之后，特斯拉停止了无线电能传输实验。后来因为摩根撤资，特斯拉陷入经济困难，华登克里夫塔被拆除抵债。尽管特斯拉的研究最终没

❶ 张卫民．我的导师尼古拉·特斯拉．[Z] http://blog.sina.com.cn/s/blog_4521e1370100dy1p.html.



图 13-2 华登克里夫塔

有结果，但是他当初的无线输电构想为后来无线供电的发展奠定了基础。

20世纪20年代中期，日本的H. Yagi和S. Uda论述了无线输电概念的可行性；30年代，美国学者开始研究不利用导线去点亮电灯的输电方案；60～70年代，Raytheon公司的William C. Brown做了大量无线供电方面的研究工作[1][2]，使得这一概念变成试验结果，奠定了现代无线供电的实验基础。

2000年以后，国际上掀起对无线电能传输技术研究与应用的热潮，无线供电技术越来越频繁地在各大通信技术展、电源新技术展上露面，各大公司纷纷推出自己的研究成果。

2001年5月，国际无线电力传输技术会议在印度洋的法属留尼旺岛召开。法国国家科学研究中心的G. Pignolet做了一个公开实验：他利用微波技术，将电能以无线方式传输，最后点亮了40m外的一只200W灯泡。

2007年6月，麻省理工大学的物理学助理教授Marin Solijacic和他的研究团队公开做了一个演示，如图13-3所示，利用电磁谐振点亮了大约2m之外连接在另一个线圈上的60W灯泡。其演示装置包括直径为3英尺的匹配铜线圈，以及与电源相连的工作频率在兆赫范围的传输线圈。接收线圈在非辐射性磁场内部发生谐振，并以相同的频率振荡，然后有效地利用磁感应来点亮灯泡[3][4]。据Marin小组称，至2008年，他们已经将传输效率提高到90%。这意味着，该技术已经具备实用化的条件。

在2008年8月的英特尔开发者论坛上，西雅图实验室的约书亚·史密斯（Joshua R. Smith）领导的研究小组再次向公众展示了这项基于“磁耦合共振”原理的无线供电技术。

---

❶ W. C. Brown. Experiments in the transportation of energy by microwave beams. [C] IEEE Intersociety Conference Record，1964，12 (2)：8.

❷ W. C. Brown. Electronic and mechanical improvement of the receiving terminal of a free-space microwave power transmission system. Raytheon Company，Wayland，MA，Tech. Report PT-4964，NASA Report No. CR-135 194，Aug. 1977.

❸ Karalis，J. D. Joannopoulos，M. Soljacic. Efficient wireless non-radiative mid-range energy transfer. Annals of Physics，Jan. 2008，323：34-48.

❹ Kurs，A. Karalis，R. Moffatt，J. D. Joannopoulos，P. Fisher，M. Soljacic. Wireless power transfer via strongly coupled magnetic resonances. Science，July 2007，317：83-86.

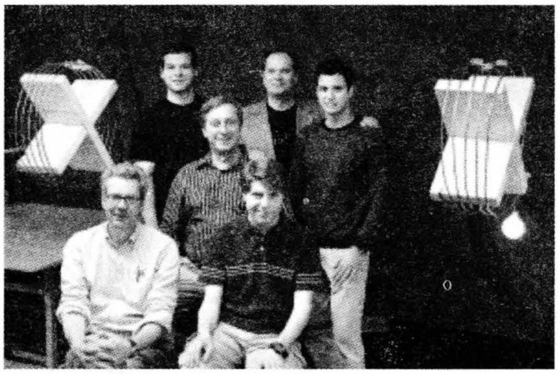


图 13-3 麻省理工大学的 Marin Solijacic 团队设计的电磁共振无线输电系统

他们在展示中成功地点亮了一只 1m 开外的 60W 灯泡，在电源和灯泡之间没有使用任何电线。在此系统中，无线电力的传输效率达到 75%，如图 13-4 所示。

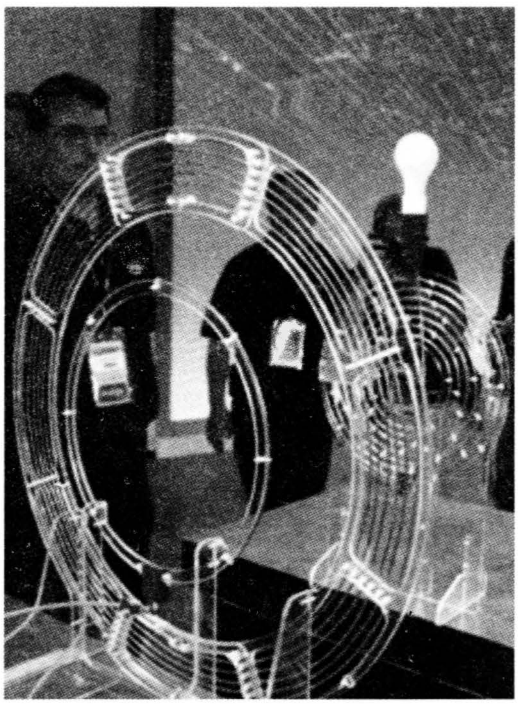


图 13-4 英特尔开发者论坛展示的无线充电技术

在 2009 年 Windows 7 发布会上，微软公司 CEO 鲍尔默带来了最新的无线视频输出和无线供电技术，把手机、PC 和笔记本等移动设备天衣无缝地结合起来，让人们感受到无线空间强大的诱惑与魅力，如图 13-5 所示。

无线充电技术源于无线供电技术。相对于大功率电能传输，小功率无线充电技术更具有使用价值。无线充电已从梦想走进现实，从概念变成商用产品。在电动牙刷、剃须刀、计步器等不少低功率日常家电产品中，无线充电技术的应用给用户带来相当的便利。特别是诺基亚 Lumia 920 配备了无线充电技术，使其得到人们的普遍关注。可以预见，无线充电功能将是旗舰智能手机的主流配置。

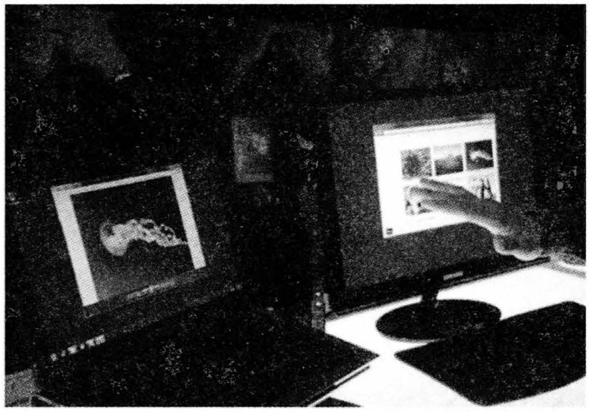


图 13-5　Windows 7 发布会上的无线屏幕输出和无线供电技术

13.1.2　无线充电工作原理分类

无线充电技术分为三类：电磁感应式、共振式和微波传输❶。

13.1.2.1　电磁感应式

目前大多数产品的无线充电功能都采用电磁感应技术。电磁感应技术主要利用经典电磁理论和变压器理论，结合现代电力自控技术，实现电能无线传输。感应式电能传输的基本原理如图 13-6 所示。在初级线圈加入交流电流 I_s，产生交变磁场强度 H，经由空气介质耦合产生磁通密度 Φ。根据法拉第电磁感应定律，次级线圈因为磁通密度 Φ 变化而感应出电动势，感应电动势 $\varepsilon = \mathrm{d}\Phi/\mathrm{d}t$。

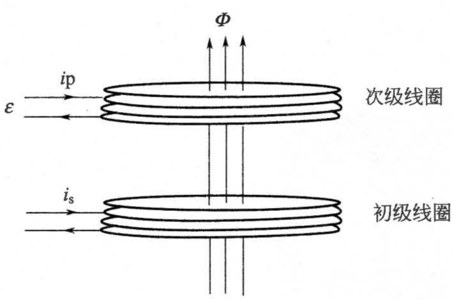


图 13-6　电磁感应基本理论

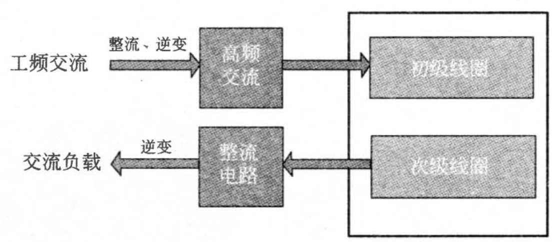


图 13-7　电磁感应原理

电磁感应技术分为感应耦合和容性耦合。其中，感应耦合的传输形式如变压器、电容等，基于铁磁芯的感应式电能传输方式在传统变压器和电机中得到广泛的应用，但是由于磁场铁芯和电场媒质的限制，它们不适合向运动的物体传输无线的隔离大气隙的能量。

如果工作频率足够高，磁场变化率将在原、副绕组之间引起很强的电磁感应，使得大气隙能量传输可行。

感应电能传输技术涉及的主要技术领域有电磁感应耦合技术、现代电力电子能量变换技术、高频磁技术、谐振逆变技术、软开关技术以及现代控制理论；具体到一个实际系统，还涉及结构设计、通信与控制技术等。感应电能传输的基本原理框图如图 13-7 所示，直接利用工频交流电作为能量供应源，可采用两相或三相工频电源，视实际的电源容量要求合理选

❶ 中国通信标准化协会. 近场无线供电技术研究，2011.

择。工频电源在经过整流电路之后向逆变电路提供平稳的直流电流。该直流电流经过逆变电路的高频逆变之后，向松耦合感应装置的初级绕组提供高频交变电流。松耦合感应装置作为感应式电能传输的关键组件，其初级绕组中通过的高频电流产生感应电磁场，并在次级绕组中产生电磁感应。在次级绕组中得到的感应电动势再通过整流或逆变后提供给直流或交流负载使用，完成非接触供电的整个能量传输过程。

另外，考虑存在多个能量接收绕组，各个绕组之间的互相影响成为关键。当某个绕组负载的等效阻抗太小（极端情况为短路）或太大（极端情况为开路）时，反应阻抗均不正常，此时将导致其他绕组负载不能工作，必须在用电设备端加上负载供电控制单元，以保证整个系统运行的稳定性和可靠性。

13.1.2.2 共振式

共振式无线电能传输的理论依据是，如果两个振荡电路具有相同的频率，在波长范围内，通过近场瞬时波耦合，感应器产生的驻波在远远小于损耗时间内，允许能量高效地从一个物体传到另一个物体。由于共振波长远远大于振荡器尺寸，所以不受附近物理的影响，而且由于磁场和生物体之间相互作用很弱，对生物体比较安全，适用于中距离传输，但是这种技术尚未达到实用化程度。

2006年10月，美国麻省理工学院教授马林·索尔贾希克（Marin Solijacic）提出了通过“磁场共振”技术实现无线电能传输的新理论，把磁场共振运用到电能传递上。他们利用铜制线圈作为电磁共振器，一组线圈附在传送电力方，另一组在接收电力方。当传送方送出某特定频率的电磁波后，经过电磁场扩散到接收方。他们成功地把一盏距离发射器2.13m的60W电灯点亮（如图13-8所示），相应的研究成果发表在2007年的《Science》杂志上。这个“隔空点灯泡”实验引起了欧美及全球各大媒体的极大关注，并进行了“Goodbye Wires”之类的广泛报道，被认为是无线电能传输技术的里程碑。他本人因为这一发明获得了麦克阿瑟基金会2008年的“天才奖”，其相应的技术被称为“WiTricity”。

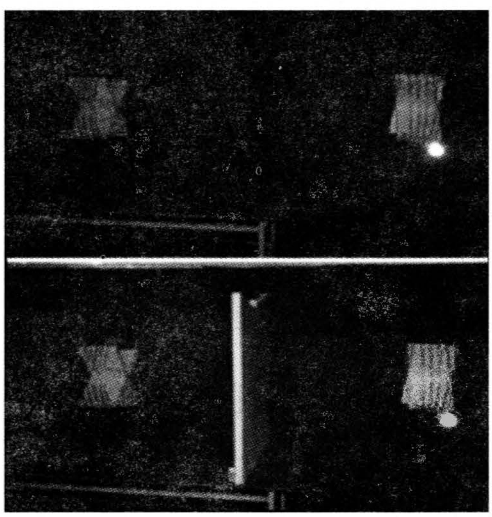


图13-8 Marin Solijacic基于磁场共振式的无线供电实验

图13-9所示是共振式无线电能传输的系统模型。A是一个半径25cm的单匝铜环，它是激励电路的一部分，输出频率为9.9MHz的正弦波；S和D是谐振线圈，B是直接连接到负载（即研究团队中实验负载为灯泡）的单匝导线环。K_S、K、K_D分别表示A和S、S和

D、D 和 B 之间的直接耦合。调整线圈 D 和 A 之间的角度，保证其直接耦合等于零，线圈 S 和 D 同轴排列，线圈 B 和 A 以及 B 和 S 的直接耦合忽略不计。系统工作在“共振”状态下(指系统中两个线圈 S 和 D 工作在相同频率共振状态)。处于谐振状态下的谐振体之间的能量交换可望达到最高的效率。如果不考虑周围空间的结构，并且干涉损耗和散失在周围环境中的损耗很低时，采用这种方法，中等距离的能量传输可以在接近全方向的状态下实现并达到很高的效率。

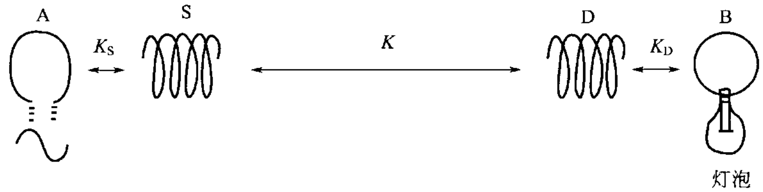


图 13-9 共振式系统模型

13.1.2.3 微波传输

基于微波辐射的无线电能传输技术（也称微波输电）将电能转化为微波能量；通过微波，自由空间中的定向辐射将能量传送到目标位置；再经过整流，转化为直流电能。这是未来电能传输的大趋势。通过微波辐射，可将太阳能转换成的电能进一步转换为微波集束能或激光能，并根据需要将束向控制在需要电能的地区，在当地再通过微波或激光接收装置将其进一步转换成电能，输入电网或直接满足不同用户的需要。微波式电能传输技术以微波输能、控制及转换原理为基础，早已在军事、科学及通信卫星等领域得到广泛应用。以美国为首的发达国家对微波输电的研究已经开展了四十多年，而在国内，这项决定未来能源走向的关键工作才刚刚起步。微波辐射式无线电能传输具有非常广泛的用途，不仅可以民用，还可以用于军事。作为太阳能发电站的传送方式，具有非常重要的研究意义。

利用微波辐射原理传输电能的系统组成如图 13-10 所示，基本步骤如下所述。

① 将直流电能转换成微波能量。

② 经波束成型天线成型发射，经空间传输。

③ 在接收终端，接收微波能量，并将其转换成直流电。

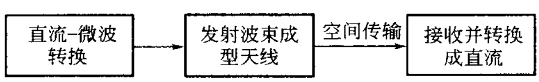


图 13-10 微波辐射传输电能系统

微波输电时，微波系统工作频率的选择是一个综合考虑的过程，需要考虑的因素包括：发射和接收天线的尺寸要求，微波元器件的效率、成本，系统对现有通信系统的影响，大气传输时的空间衰减等。微波通信使用的频率大部分在 2～20GHz 范围内，所以对于不同的地形条件，其反射系数及电平损耗是不同的。

在接收端，微波输电接收单元的关键因素是接收整流天线。这种天线包括天线和整流器的功能，能够实现能量收集、谐波抑制和整流。

微波输电的总效率等于直流到微波、微波传输以及接收整流三部分效率的总乘积。至 2008 年，直流转换微波能量装置的效率为 70%～90%；发射天线的效率为 70%～97%；空间传输的效率为 5%～95%，取决于具体情况；接收转换直流的效率为 85%～92%。尽管各部分在各自的相关实验中分别能达到最大效率，却不能在一个完整的系统中同时实现各自的

最大值。现已被实验证实的最大总效率为 54%❶。如果能将各个部分的传输效率更好地匹配，通过改进接收整流天线的设计，尤其是采用特殊设计的二极管，提高二极管以及接收天线的效率等措施，总传输效率的理论最大值可达 76%❷。

13.1.2.4 三种无线充电技术的比较

在介绍完三种无线充电技术后，对这三种无线充电技术做一个比较，如表 13-1 所示。

表 13-1 三种充电技术比较

技术类型			基本理论	适用功率	传输距离	典型应用
感应式	感应耦合	紧耦合	磁路、交流电路	各种功率	近	变压器
		松耦合	分布式磁场	中低功率		感应式电能传输系统
	容性耦合	紧耦合	电场	低功率		电容
共振式	磁场共振		分布式电场、磁场	中低功率	中	WiTricity 技术
微波式	微波辐射		电磁波远场	中高功率	远	空间太阳能发电输电

13.1.3 无线充电标准介绍

目前，无线充电技术不算成熟，标准尚未统一。主流的无线充电标准主要有三种：Power Matters Alliance（PMA）标准、Qi 标准和 Alliance for Wireless Power（A4WP）标准❸。另外，NFC Forum 也在制定基于 NFC 技术的无线充电标准，标准化工作正在进行中。本节将详细介绍这 4 种无线充电标准。

13.1.3.1 Qi

目前全球最主要的无线充电标准是 Qi 标准，由全球首个推动无线充电技术的标准化组织无线充电联盟（Wireless Power Consortium）推出。WPC 联盟成立于 2008 年，其目的是达成无线充电技术标准的统一，并确保任何成员公司之间产品的兼容性。WPC 联盟的成员超过 100 多家，包括 Verizon、NTT Docomo、诺基亚、三星、德州仪器、华为等公司，覆盖无线充电解决方案各生产环节。Qi 采用目前最为主流的电磁感应技术，目的是为所有电子厂商提供统一的通用标准。

Qi 标准支持的充电功率为 0～5W，通过在两个平面线圈之间使用电感耦合，将电力从电力发送器传输给电力接收器。两个线圈之间的距离一般为 5mm。输出电压调节由一个全局数字控制环路负责，电力接收器与电力发送器通信，进行充电参数协商。如图 13-11 所示，Qi 标准定义了电力发送器、电力接收器以及发送器和接收器之间的通信协议❹。

电力传输方向始终是从电力发送器到电力接收器。电力发送器的关键电路是用于向电力接收器传输电力的一次线圈、驱动一次线圈的控制单元以及解调一次线圈电压或者电流的通信电路。电力接收器通常为一种便携式设备。电力接收器的关键电路是用于从电力发送器接收电力的次级线圈、用于将交流转换为直流整流电路、用于将未稳压直流转换为经过稳压的直流电源调节电路以及用于将信号调制到次级线圈的通信电路。

---

❶ 张猛．微波无线电能传输与通信系统的研究［D］．浙江大学学位论文，2004.

❷ Mankins J. C. A fresh look at the concept of space solar power. Proceedings of SPS' 97 conference：S7041. Montreal，Canada，1999.

❸ 张召林．无线充电技术的三大主流标准：PLA、Qi 和 A4WP. http：//www.leiphone.com/1113-zzl-pla-qi-a4wp.html.

❹ Wireless Power Consortium. http：//www.wirelesspowerconsortium.com/.

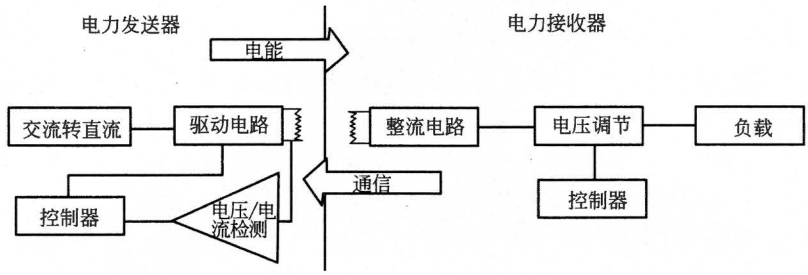


图 13-11　Qi 无线充电功能结构图

通信协议包括模拟和数字声脉冲（pinging）信号、身份识别和配置以及电力传输。大致的通信机制如下所述：

① 来自电力发送器的模拟 ping 信号检测到对象存在。

② 来自电力发送器的数字 ping 信号为模拟 ping 信号的加长版，并让电力接收器有时间回复一个信号强度包。如果该信息强度包有效，电力发送器会让线圈保持通电，并执行下一步骤。

③ 身份识别和配置阶段期间，电力接收器发送一些数据包，对其进行身份识别，并向电力发送器提供配置和设置信息。

④ 在电力传输阶段，电力接收器向电力发送器发送控制误差包，以增加或者减少功率。正常运行期间，每隔约 250ms 便发送这些包。另外，在正常运行期间，电力发送器每隔 5s 发送一次电力包。

⑤ 为了终止电力传输，电力接收器会发送一条“终止充电”消息，或者 1.25s 时长内都不通信。两种事件中的任何一个都会让电力发送器进入低功耗状态。

Qi 标准的典型代表性产品有诺基亚 Lumia 920（如图 13-12 所示）、诺基亚 Lumia 820、谷歌 Nexus 4 等。使用这些手机时，不需要安装任何配件，直接将它放在任何一款支持 Qi 标准的充电器上就能充电。目前市面上有劲量、PowerMate 等品牌的大量不同款式的无线充电器可供选择。

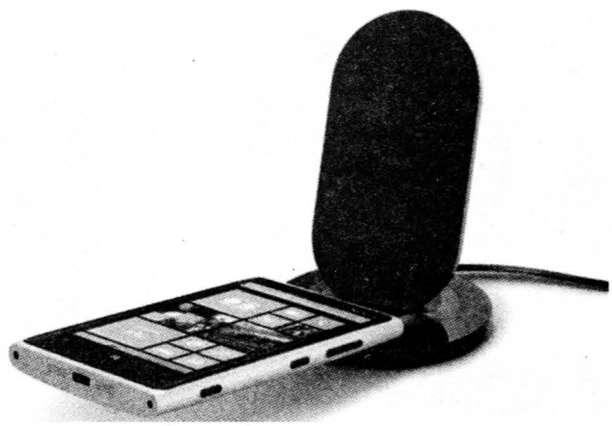


图 13-12　诺基亚 Lumia 920

13.1.3.2　A4WP

A4WP（Alliance for Wireless Power）是由高通公司、三星公司以及 Powermat 公司共同创建的无线充电联盟。该联盟还包括 Ever Win Industries、Gill Industries、PeikerAcustic 和 SK Telecom 等成员（如图 13-13 所示），目标是为包括便携式电子产品和电动汽车等在内

的电子产品无线充电设备建立技术标准[1]。

A4WP采用电磁共振无线充电技术，与Qi的电磁感应技术有所区别。这两种技术各有千秋。前者传输效率可能较低，但可以实现稍远距离的无线充电；后者需要近距离接触，例如将手机放在底座上，不用接线就可以通过感应充电，充电效率较高。

A4WP标准组成联盟，希望让无线充电迅速普及，让用户在任何地方都可完成无线充电。另外，A4WP扩大了产品覆盖范围，包括电动汽车等更广泛的电子产品领域。由于联盟组成时间较晚，目前采用该标准的产品并不多。

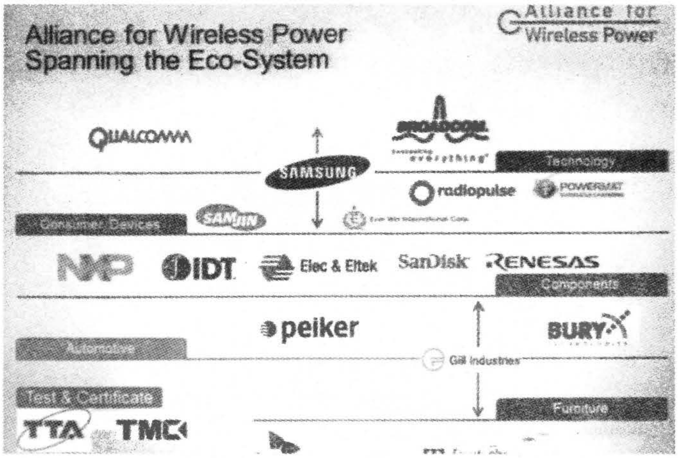


图13-13 A4WP联盟

13.1.3.3 PMA

Power Matters Alliance标准是由Duracell Powermat公司发起的，该公司由宝洁公司与无线充电技术公司Powermat合资经营，拥有比较出色的综合实力。除此以外，Powermat是Alliance for Wireless Power（A4WP）标准的支持成员之一。

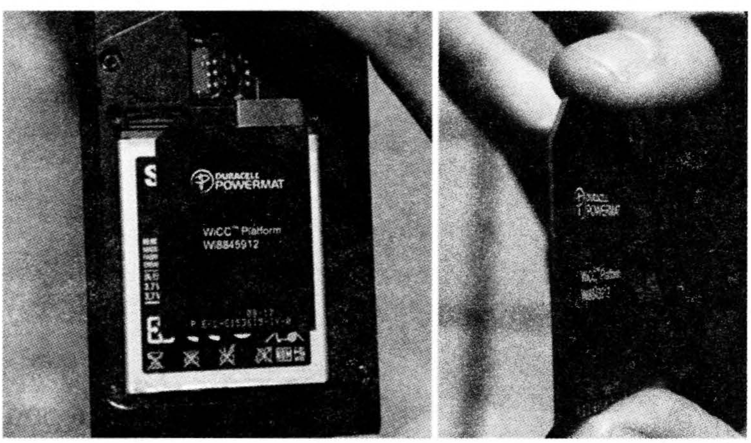


图13-14 PMA WiCC充电卡

目前有AT&T、Google和星巴克三家公司加入PMA联盟。PMA联盟致力于为符合IEEE标准的手机和电子设备打造无线供电标准，在无线充电领域中具有领导地位。

---

[1] Alliance for Wireless Power. http：//www. a4wp. org/.

Duracell Powermat 公司推出过一款 WiCC 充电卡，采用的就是 Power Matters Alliance 标准。WiCC 比 SD 卡大一圈，如图 13-14 所示，其内部嵌入了用于电磁感应式非接触充电的线圈和电极等组件，卡片厚度较薄，插入现有智能手机电池旁边即可使用。利用该卡片，可使很多便携终端轻松支持非接触充电。

13.1.3.4 NFC Forum

NFC 技术除了应用在手机支付、数据传输等领域外，还可以实现无线充电。NFC 数据的传输基于电磁感应原理，因此 NFC 天然地具有无线充电可能。2011 年 11 月，在 LG、索尼、NEC、松下、瑞萨、VTT、意法半导体等公司推动下，NFC Forum 成立了无线充电工作组，制定基于 NFC 无线充电技术的标准[1]。目前，NFC Forum 无线充电工作组已经完成了需求收集，正在制定 NFC 无线充电控制流程及电能传输规范。

根据充电功率的大小，有三种 NFC 无线充电应用场景。

（1）低功率 NFC 无线充电场景

主要应用于具有 NFC 功能的计步器、蓝牙耳机、MP3 播放器等低功率电子产品的无线充电。充电器可以是具有 NFC 功能的笔记本电脑等能够提供高功率的电子产品，如图 13-15 所示。应用场景是：将计步器放在笔记本电脑上，通过 NFC 传输数据，同时笔记本电脑对计步器充电。

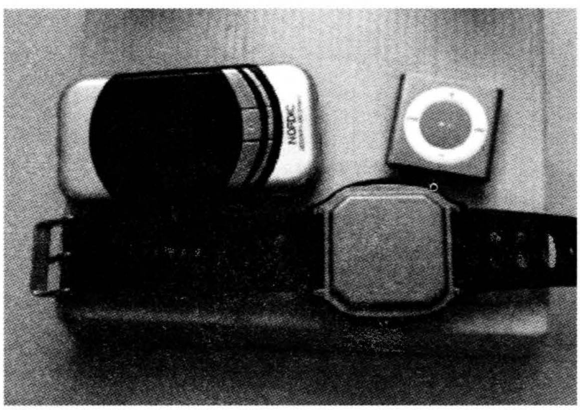


图 13-15 低功率充电场景

（2）中功率 NFC 无线充电场景

主要应用于具有 NFC 功能的智能手机、照相机等电子产品的无线充电。充电器是具有 NFC 功能的笔记本电脑、充电板等能够提供高功率的电子产品，如图 13-16 所示。应用场景是：将智能手机放置在笔记本电脑上，在通过 NFC 传输图片的过程中，笔记本电脑为智能手机无线充电。

（3）高功率 NFC 无线充电场景

主要应用于具有 NFC 功能的单反相机、平板电脑等高功率电子产品的无线充电。充电器可以是接有电源的无线充电板。比如，将单反相机或平板电脑放置在 NFC 无线充电板上充电，如图 13-17 所示。

尽管 NFC 无线充电有三种应用场景，但是就目前来说，由于技术的限制，NFC 无线充电主要用于对中低功率电子产品的充电场景。表 13-2 列出了 NFC 设备无线充电的功率要求。

---

[1] NFC Forum. www.nfc-forum.org.



图 13-16　中功率充电场景



图 13-17　高功率充电场景

表 13-2　功率要求

NFC设备	电池容量	慢(10h)	中(4h)	快(2h)
耳机	300mA·h (4V)	130mW	300mW	600mW
智能手机	1000mA·h (4V)	400mW	1W	2W
相机	1200mA·h (4V)	480mW	1.2W	2.4W
相机	1500mA·h (8V)	1.4W	3.6W	7.2W
笔记本电脑	7000mA·h (12V)	8W	20W	40W

与Qi充电技术相比，NFC无线充电技术主要定位于0～5W的充电场景，目前以0～1W为主。Qi主要应用于1W以上的设备，如图13-18所示。Qi的工作频率为110～220kHz，因此需要较大的天线，不适合应用于体积较小的低功率电子产品中，如计步器、蓝牙耳机等。所以，NFC无线充电技术将是Qi技术的有益补充。

随着NFC技术的普及，越来越多的电子产品集成了NFC功能，也为NFC无线充电技术的推广提供了较好的市场基础。在可见的未来，所有带有NFC功能的台式机、笔记本电脑都可以为手机、PAD、无线鼠标键盘等设备无线充电；其次，由于整个充电功能及过程由NFC芯片控制，通过NFC无线充电控制协议可以在充电过程中实现充电控制、用户认证、状态切换、不同充电技术的选择、切换等功能，提高了无线充电效率。另外，NFC无线充电与NFC数据传输共用一个天线，不仅减小了NFC设备尺寸，而且可以降低NFC设备成本。最后，可以将NFC无线充电的其他应用结合在一起，挖掘出新的商业模式，如将

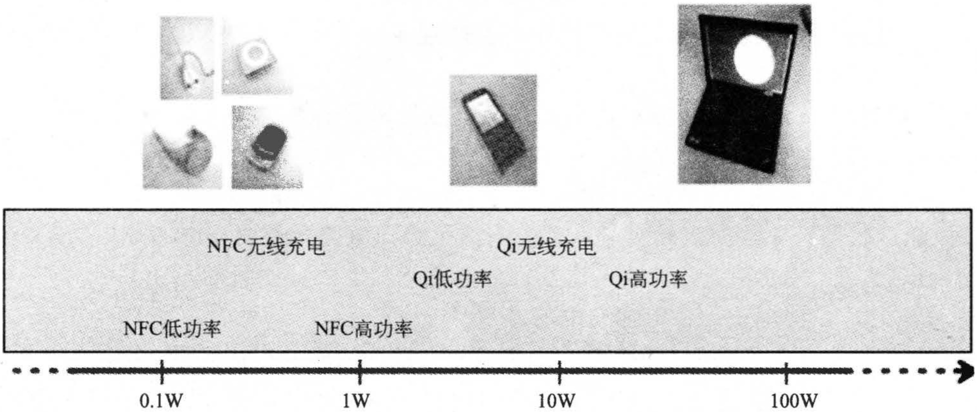


图 13-18 NFC 无线充电与 WPC 无线充电

无线充电与移动支付相结合。

13.2 NFC 无线充电系统架构

NFC 工作频率为 13.56MHz，在数据传输的过程中伴随着能量输出，因此在提高功率的前提下，一个 NFC 设备可以为另一个 NFC 设备充电；在保证数据传输的情况下，提高输出功率进行无线充电。另外，NFC 作为一种数据通信方式，也可以用于其他无线充电技术的控制流程，如使用 NFC 技术来选择相应的无线充电技术、交互充电参数等。这两种 NFC 无线充电场景分别对应两种不同的 NFC 无线充电系统架构：共用 NFC 天线的无线充电系统和 NFC 协作式无线充电系统。下面将详细介绍这两种系统架构。

13.2.1 共用 NFC 天线的无线充电系统

共用 NFC 天线的无线充电系统指无线电能的收发通过 NFC 天线来完成，其系统架构如图 13-19 所示。

电能发送器包括以下 3 个部分：

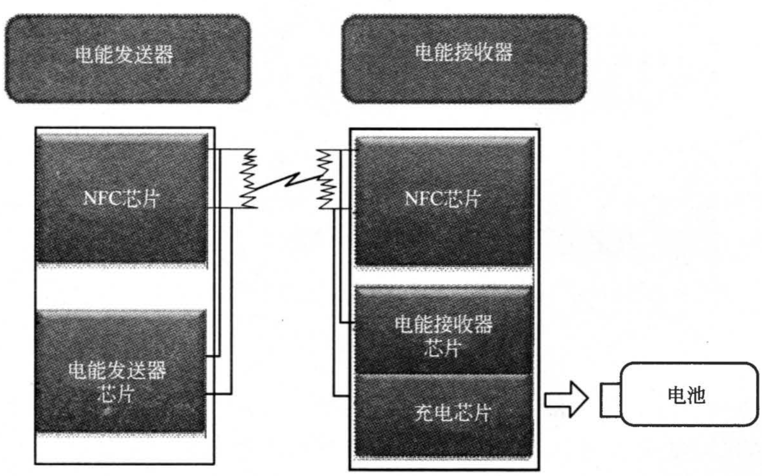


图 13-19 共用 NFC 天线的无线充电系统

① NFC 芯片：实现 NFC 协议通信以及无线充电控制流程的芯片。

② 电能发送器：实现电能传输的芯片，包括直流转交流、整流、逆变，以及驱动 NFC 天线的控制单元。

③ NFC 天线：实现 NFC 数据传输及作为电能发送器的发送线圈。

电能接收器包括以下 4 个部分：

① NFC 芯片：实现 NFC 协议通信以及无线充电控制流程的芯片。

②电能接收器：实现交流转直流、整流电路及稳压电路，并向充电芯片提供输入电压。

③ NFC 天线：实现 NFC 数据传输及作为电能接收器的接收线圈。

④ 充电芯片：向电子设备电池充电的芯片。

图 13-20 举例说明了共用 NFC 天线的无线充电系统架构的应用场景。

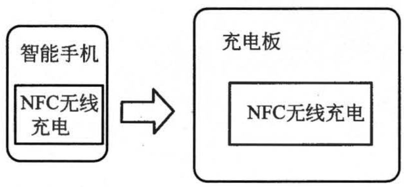


图 13-20　共用 NFC 天线应用场景

如图中所示，有两个设备，一个是具有 NFC 无线充电功能的智能手机，另一个是具有 NFC 无线充电功能的充电板。这里，充电板为电能的发送器，智能手机为电能接收器，即充电板为智能手机充电。当智能手机靠近充电板时，充电板检测到智能手机，建立 NFC 的通信链路并进行充电参数协商。参数协商完毕，即传输电能。在电能传输的过程中，也可以交换 NFC 数据；如果该充电板位于笔记本电脑上，智能手机可以在充电过程中通过 NFC 将手机中的图片传递给笔记本电脑。

13.2.2 NFC 协作式无线充电系统

NFC 协作式无线充电系统是指 NFC 通信只作为其他无线充电技术的控制流程，如无线充电技术的选择、充电参数的交换、充电的起始控制等；电能传输通过不同的无线充电技术来完成，比如 Qi、A4WP、PMA 等。图 13-21 描述了 NFC 协作式无线充电系统的架构。

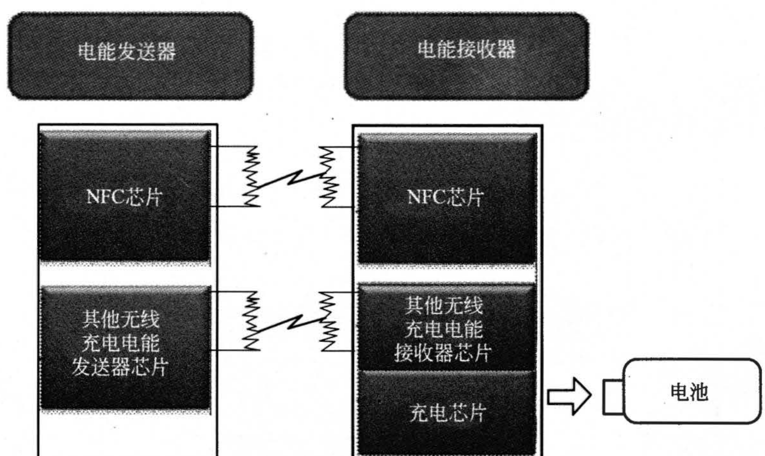


图 13-21　NFC 协作式无线充电系统的架构

电能发送器包括以下 4 个部分：

① NFC 芯片：实现 NFC 协议通信以及无线充电控制流程的芯片。

② NFC 天线：实现 NFC 数据传输的天线。

③ 电能发送器：实现某个无线充电技术的电力传输芯片，包括直流转交流、整流、逆

变以及驱动发送天线的控制单元。

④ 电能发送器天线：负责电能传输的线圈。

电能接收器包括以下 4 个部分：

① NFC 芯片：实现 NFC 协议通信以及无线充电控制流程的芯片。

② NFC 天线：实现 NFC 数据传输。

③ 电能接收器：实现交流转直流、整流电路及稳压电路，并向充电芯片提供输入电压。

④ 电能接收器芯片：负责电能接收的线圈。

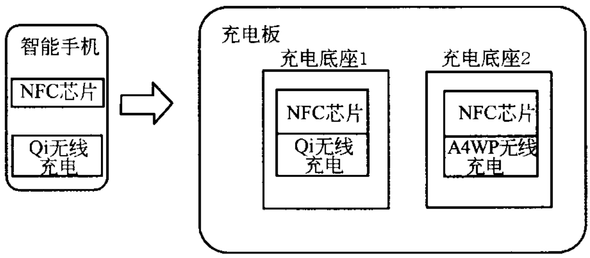


图 13-22 NFC 协作式无线充电系统架构的应用场景

图 13-22 举例说明了 NFC 协作式无线充电系统架构的应用场景。从图中看到，有两个设备，一个是具有 NFC 通信功能及 WPC 技术的智能手机，另一个是具有 NFC 通信功能及 Qi 技术、A4WP 技术的充电板。当智能手机靠近充电板时，充电板与智能手机建立 NFC 通信链路，并进行充电技术协商。智能手机通过 NFC 链路向充电板说明其支持 Qi 的无线充电技术。由于充电板也支持 Qi 技术，因此充电板和智能手机进一步通过 NFC 通信进行充电参数协商。参数协商完毕后，通过 Qi 技术进行无线充电。由于 Qi 本身也定义了充电的控制流程，因此 NFC 可以只作为充电技术的选择。当然，NFC 也可以为 Qi、A4WP 等充电技术定义统一的控制流程。

13.3 NFC 无线充电控制协议

13.3.1 无线充电流程

NFC 无线充电主要包括两个部分：控制流程和无线电能传输。控制流程通过 NFC 通信来完成无线充电技术的选择、无线充电参数的协商、用户认证、充电起始控制等控制流程。无线电能传输的工作集中在共用 NFC 天线的无线电能传输上，而采用其他无线电能传输技术的定义不在工作组工作范围内。当然，工作组希望能够在控制流程上兼容共用 NFC 天线系统和 NFC 协作式无线充电系统。图 13-23 描述了 NFC 无线充电控制流程与无线电能传输的过程。

NFC 无线充电有两种充电流程。第一种如图 13-24 所示。

具有 NFC 无线充电功能的电能发送方进行周期性的轮询来发现靠近的 NFC 设备。当一个具有 NFC 无线充电功能的电能接收方靠近时，发送方进行设备发现并与接收方建立 NFC 链路。发送方开启无线充电控制流程，获取接收方相关电能/电量参数，并执行参数协商过程；然后，电能发送方的 NFC 芯片通知电能发送芯片传输电能。之后进行第一轮无线电能传输。此时，NFC 通信链路将被释放。第一轮无线电能传输完毕后，发送方与接收方重新建立通信链路并进行电能/电量相关参数的协商。如果需要继续充电，发送方将发起第二轮

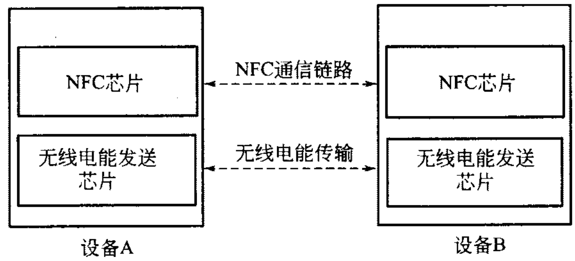


图 13-23 控制流程及电能传输

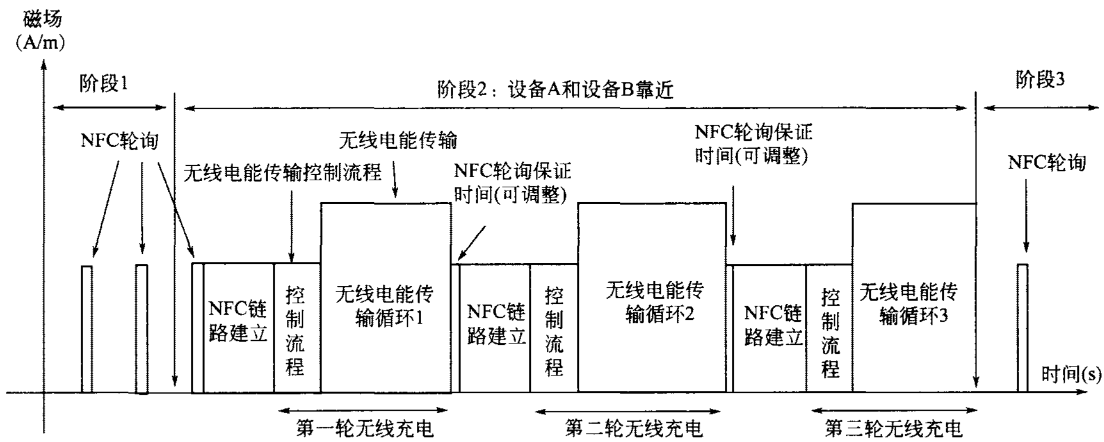


图 13-24 第一类 NFC 无线充电流程

无线电能传输过程。NFC 链路建立、控制流程及电能传输过程周期进行，直到接收方告知发送方电量已充满，无线充电流程完毕。

第二种 NFC 无线充电流程如图 13-25 所示。

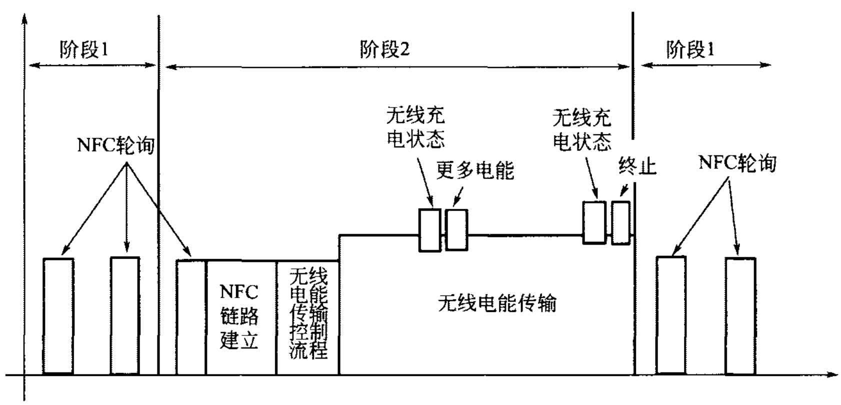


图 13-25 第二类 NFC 无线充电流程

具有 NFC 无线充电功能的电能发送方进行周期性的轮询来发现靠近的 NFC 设备。当一个具有 NFC 无线充电功能的电能接收方靠近时，发送方进行设备发现，并与接收方建立 NFC 链路。发送方开启无线充电控制流程，获取接收方相关电能/电量参数，并执行参数协商过程。协商完毕后，电能发送方的 NFC 芯片通知电能发送芯片传输电能。在电能传输过程中，保持 NFC 通信链路，并周期地交换充电状态，确认接收方是否需要继续充电。当接

收方确认充电完成后，发送方结束充电过程。

可以看出，第一类和第二类无线充电流程的主要区别在于电能传输与控制流程是否交替进行。在第一类充电流程中，充电控制流程与电能传输交替进行，每一个循环都会执行一次完整的控制流程；在第二类充电流程中，无线充电控制流程之后传输电能。在电能传输的过程中周期地交换充电状态。

这两种方式在工作组中都有支持者。目前，这两种方案趋于融合，在第一类充电流程的基础上，在充电控制流程中加入了安全认证，并在后续周期的控制流程中只进行充电状态的交互而不重新开启一轮新的控制流程，如图 13-26 所示。

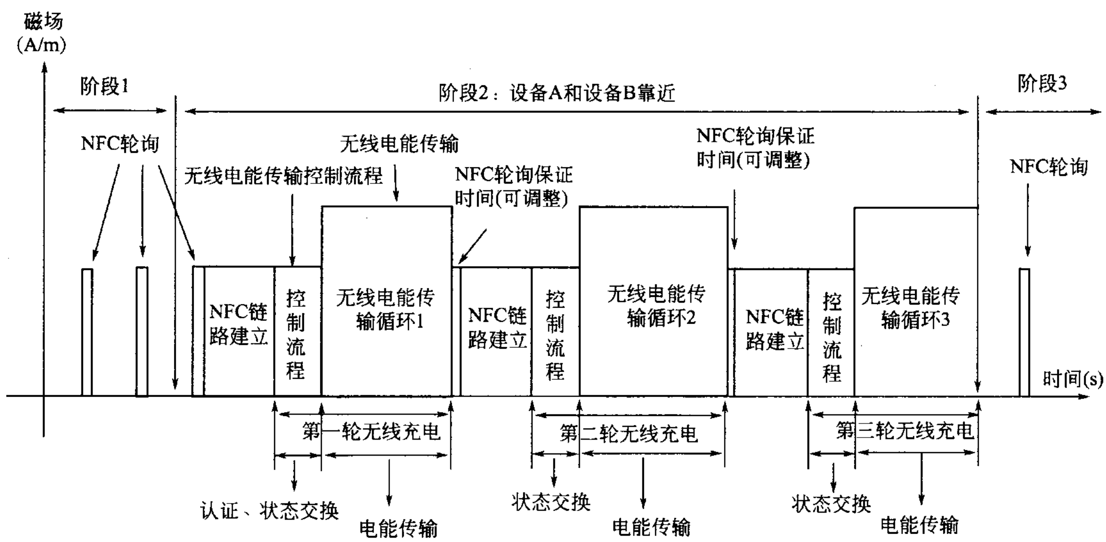


图 13-26　第一类充电流程与第二类充电流程的融合

13.3.2　控制流程

尽管有两种无线充电系统架构及两类无线充电流程，但是对于控制流程，有一个统一的 NFC 无线充电控制流程来兼容不同的系统架构及充电流程。

当电能接收方靠近发送方时，发送方将开启 NFC 无线充电的控制流程，如图 13-27 所示。电能发送方控制流程如图 13-28 所示，电能接收方控制流程如图 13-29 所示。

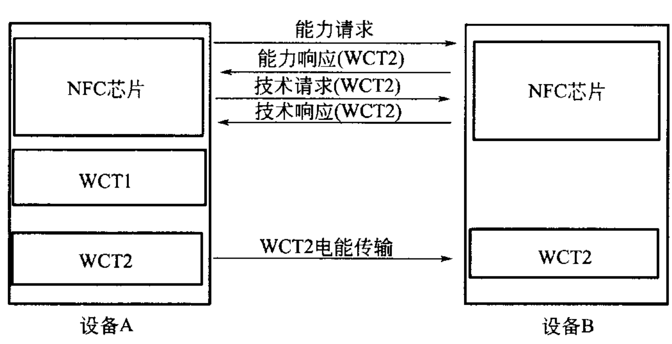


图 13-27 NFC 无线充电控制流程

详细的无线充电控制流程步骤如下所述：

① 电能发送方发送能力请求帧（capability request frame）。该帧主要用于发送方获取接

收方所支持的无线充电技术列表。

② 电能接收方向发送方响应能力响应帧（capability response frame)。该帧用于接收方向发送方通告自己所支持的无线充电技术列表。接收方支持一个或多个不同的无线充电技术，如 NFC、QI、A4WP、PMA 等。

③ 电能发送方在收到能力响应帧后，判断自己与接收方是否有共同支持的无线充电技术。如果有，电能发送方向接收方发送技术信息请求帧（technical information request frame)，该帧用于电能发送方选择某一个无线充电技术用于接下来的电能传输；如果电能发送方和接收方没有共同支持的无线充电技术，充电流程结束。

④ 电能接收方收到技术信息请求帧后，向发送方响应技术信息响应帧（technical information response frame)，用于确认是否采用该无线充电技术传输电能。

⑤ 电能发送方收到技术信息响应帧后，开始传输电能。

⑥ 电能传输完毕后，如果接收方不再需要继续充电，发送方和接收方交互传输终止帧（power transmission finalization frame）来结束无线充电过程。如果接收方还需要继续充电，对于第一类充电流程，继续执行步骤①；对于第二类设备，继续执行无线电能传输。

在无线充电过程中，只要电能接收方离开发送方，电能传输将会终止。对于共用 NFC 天线的无线充电系统，在控制流程执行完毕后，将会执行无线电能传输；对于 NFC 协作式无线充电系统，电能发送方和接收方交换技术信息请求/响应帧，确定采用除 NFC 无线充电技术之外的充电技术，如 QI、A4WP、PMA 等。

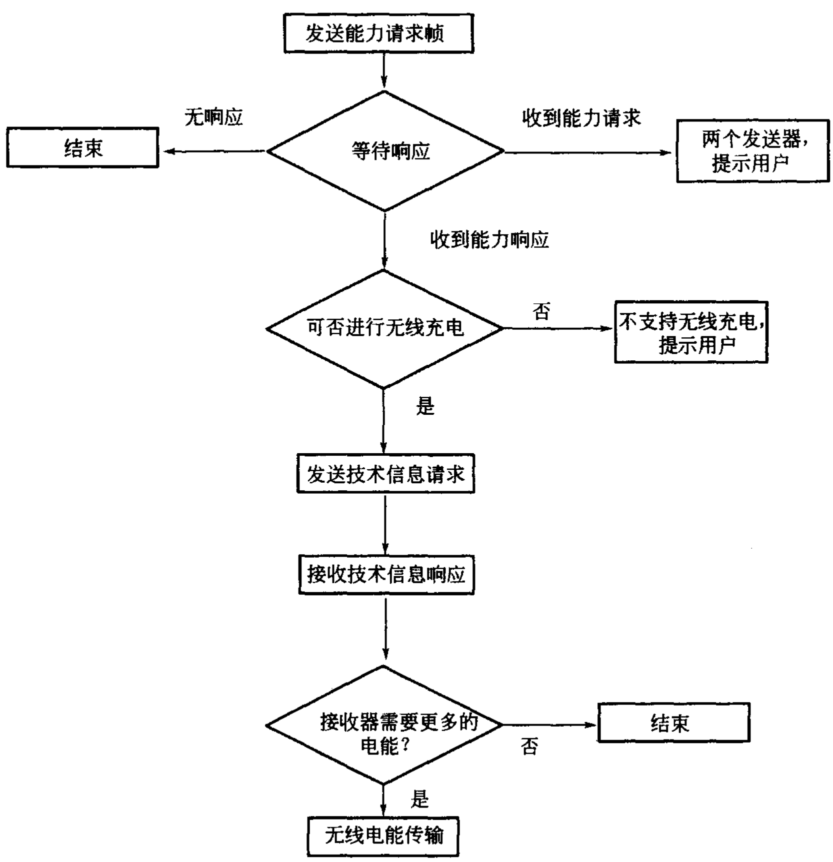


图 13-28　电能发送方控制流程

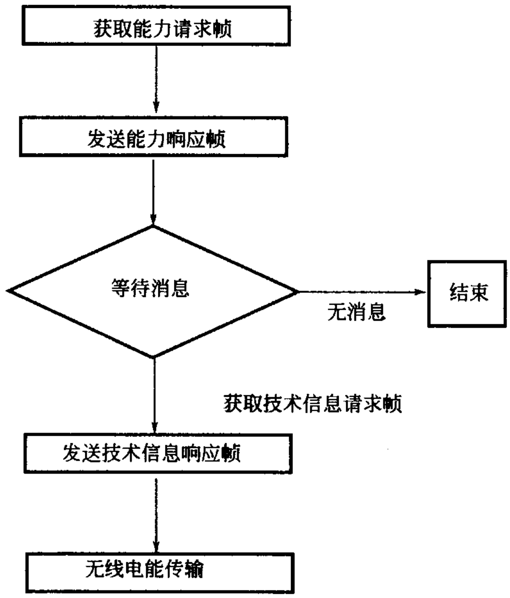


图 13-29　电能接收方控制流程

13.3.3　控制帧

（1）能力请求帧（capability request frame）

能力请求帧用于电能发送方获取接收方所支持的无线充电技术列表。该帧中还包括关于是否使用 NFC 通信作为控制流程，以及 NFC 无线充电控制协议的版本号等信息。

（2）能力响应帧（capability response frame）

电能接收方收到能力请求帧后，向发送方响应自己支持的无线充电技术列表，如接收方支持一个或多个不同的无线充电技术，如 NFC、QI、A4WP、PMA 等；还包括以下内容：

① NFC 无线充电控制协议版本信息。

② 电能接收方支持无线充电技术的个数。

③ 所支持的无线充电技术的名称。

④ 无线充电技术的工作频率。

⑤ 是否支持 NFC 通信作为控制流程。

⑥ 电能接收方设备相关信息，如设备名称、制造商等。

⑦ 电能接收方支持的功率等级，如低功率、中等功率、高功率。

当电能发送方收到能力响应帧后，进行判断。遇到以下两种情况中的任何一种，即可结束充电流程。

① 没有共同支持的无线充电技术。如发送方支持 NFC、QI，但是接收方只支持 A4WP。

② 双方中有一方或都不支持 NFC 通信作为控制流程。注意，这里是指 NFC 无线充电流程结束，不包括其他无线通信技术通过自己的控制流程进行无线充电。

（3）技术信息请求帧（technical information request frame）

技术信息请求帧用于电能发送方选择某一个无线充电技术用于接下来的电能传输。该帧包括以下内容。

① 所选择的无线充电技术的名称。

② 所选择的无线充电技术的工作频率。

③ 电能传输的持续时间，即图 13-24 和图 13-26 中表述的电能传输的一个周期。

(4) 技术信息响应帧（technical information response frame）

技术信息响应帧用于电能接收方确认是否采用技术信息请求帧中描述的无线充电技术传输电能。该帧包括以下内容：

① 确认选择的无线充电技术的名称。

② 确认选择的无线充电技术的工作频率。

③ 最大允许的磁场强度，用于电能发送方限制最大的功率输出。

④ 是否需要更多电量。如果是，发送方继续传输电能；如果否，发送方停止传输电能。

⑤ 剩余电量信息（百分比）。

⑥ 下一个电能传输循环的持续时间。

⑦ 电能传输停止原因，用于描述为什么接收方要停止电能传输，如出现错误、过热、电量已充满等原因。

⑧ 电能接收方的温度。

13.3.4 控制流程的通信模式

对于控制流程的通信方式，目前有两种不同的观点：读卡器模式和点对点（P2P）模式。

(1) 读卡器模式的控制流程

读卡器模式是指电能发送方工作在读卡器模式，而电能接收方工作在卡模拟模式。读卡器模式的控制流程支持目前 NFC Forum 定义的四种标签格式的读写。具体通信流程如图 13-30 所示。

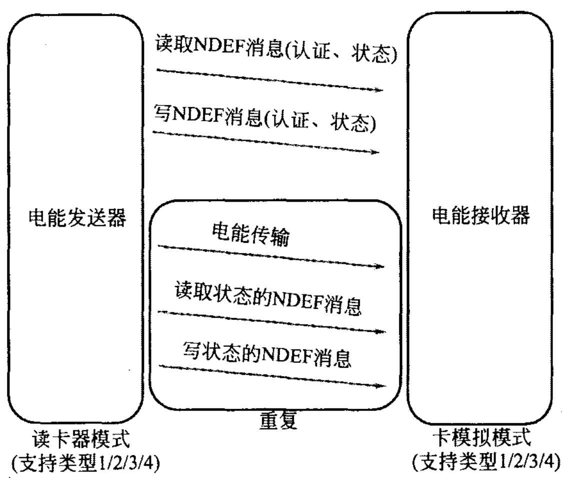


图 13-30 基于读卡器模式的控制流程

① 电能发送方工作在读卡器模式，从工作在卡模拟模式下的电能接收方处读取相关的认证信息及电能状态信息。这些信息被封装在 NDEF 消息中。

② 电能发送方根据读取的内容，向接收方写入相关的确认信息。

③ 电能传输开始。

从整体流程上看，读卡器模式的控制流程与前面介绍的控制流程并无矛盾之处，因为读卡器模式的通信方式总是由读卡器发起的读写命令完成的。NDEF 消息中包含能力响应帧及

技术信息请求帧。

(2) 点对点模式的控制流程

点对点模式的控制流程是指电能发送方和接收方工作在点对点模式下，通过 NDEF 消息交换控制帧。

通过分析可以看出，这两种通信模式各有优缺点。读卡器模式控制流程最大的优点在于如果接收方的电池处于完全馈电的状况，接收方可以从发送方处获取能量来交换 NDEF 消息；而且对于电能接收方，无需集成 NFC 芯片，只需要一个 NFC 标签就可以完成控制流程，因此从成本上、设备尺寸上来看更有优势。点对点模式控制流程最大的优点在于点对点通信本身具有很大的灵活性，但是由于点对点通信采用 LLCP 协议作为承载，一般情况下该协议运行在电子设备的主控处理器上；如果电池已没有电，基于点对点模式的控制协议无法运行，这也是点对点模式控制流程的最大问题。

13.4 小结

本章介绍了无线充电的发展历史、各种无线充电技术（如 Qi、A4WP、PMA、NFC）的工作原理及相关标准化组织情况，着重阐述了基于 NFC 技术的无线充电系统架构、统一的控制流程等内容。虽然 NFC 无线充电规范还未发布，但是 NFC 无线充电技术吸引了各大厂商积极参与。由此可见，随着 NFC 技术的普及、NFC 无线充电技术的成熟，NFC 无线充电将有非常好的市场前景。

第14章 NFC移动支付

14.1 移动支付概述

14.1.1 移动支付的概念

对于移动支付来说，没有统一的概念，目前普遍认为移动支付是通过移动终端（如手机）对所购买的商品进行支付的一种方式。这个定义只是涵盖了移动支付生态系统中的一个方面。实际上，移动支付有一个完整的生态链，从生态链上不同的环节出发，会得出关于移动支付不同的概念。图 14-1 所示是 Mobile Payments Today[1] 对于移动支付的描述，它也没有给出统一的、标准的移动支付定义，但是从不同角度描述了移动支付可以做什么。

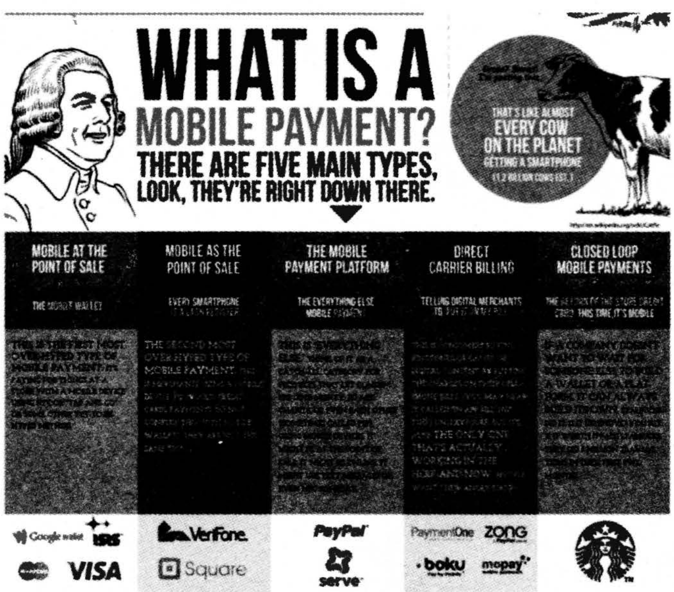


图 14-1 什么是移动支付

[1] http：//www. mobilepaymentstoday. com/.

从图中可以看出，Moible Payments Today 将移动支付分为 5 个类型，如下所述。

（1）移动钱包

移动钱包即前面提到的对于移动支付的普遍定义。将移动终端作为一个“钱包”，通过手机刷卡乘坐公交车、地铁，通过手机刷卡来购买商品等，如谷歌钱包（Google Wallet）。

（2）移动 POS 机

将移动终端作为一台 POS（Point Of Sale，销售终端）机，主要用于商家。如国外的 Square，国内的盒子支付等，它们通过在手机上外接一个刷卡器，将手机作为一台 POS 机，接受银行卡支付。

（3）移动支付平台

上面两点提到的是移动支付终端：移动钱包和移动 POS 机。移动支付平台是指除了这两个终端之外的所有相关内容，如移动 POS 机要接入的银行收单系统，以及接入收单系统所需的无线网络（GPRS/3G/WiFi）。移动钱包还可以接入第三方移动支付服务提供商，如支付宝、Paypal 等。所有这些移动支付生态系统中的环节都是移动支付平台的一部分。

（4）基于电信运营商的移动支付

Mobile Payments Today 还给出了基于电信运营商的移动支付类型。这种类型的支付在早些年非常常见，目前国外的某些电信运营商依然采取这种方式实现移动支付，即通过通信费购买商品，如购买彩铃、游戏等增值服务。由于通信费直接由运营商收取，因此这种类型的移动支付不需要银行介入。

（5）闭环的移动支付

闭环的移动支付通常用在某一个特定的商家，商家为客户提供自己的移动支付方式，比如星巴克。2011 年，星巴克推出了“星巴克数字卡”应用。该应用安装在 iPhone 和 Android 系统上，消费者可以对该数字卡充值，然后在收银处通过扫描二维条码完成支付。

从 Mobile Payments Today 对于移动支付的分类可以看出，移动支付是一个完整的生态圈，它包含与支付相关的各个环节。以中国移动支付为例，图 14-2 列出其生态圈的参与方。

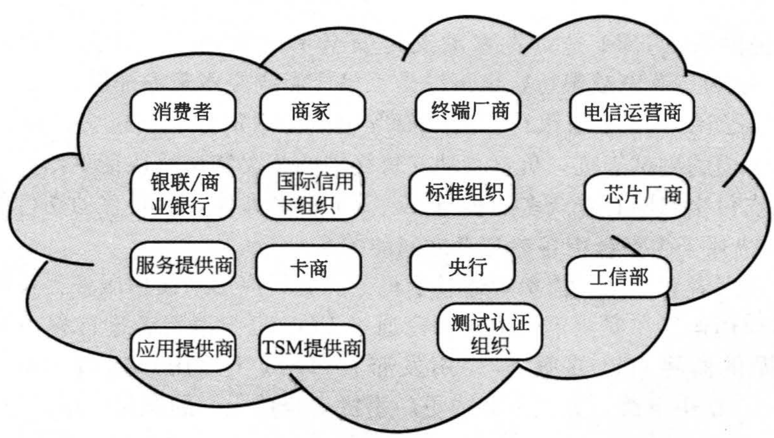


图 14-2 移动支付生态圈

① 消费者：商品或服务的购买者。消费者通过移动终端上的移动钱包或支付应用完成移动支付。

② 商家：商品或服务的提供方，与消费者交易。

③ 终端厂商：包括手机终端厂商和 POS 机厂商，分别为消费者和商家的移动支付交易提供终端支持。

④ 电信运营商：为消费者和商家提供无线或有线网络服务。电信运营商也可以作为服务提供商、TSM（Trusted Service Management，可信服务管理）提供商、测试认证、标准制定者、应用提供商等。

⑤ 银联/商业银行：为消费者提供个人账户管理、交易流程管理等金融相关服务，也可以作为 TSM 提供商、服务提供商、测试认证、标准制定者、应用提供商等。

⑥ 国际信用卡组织：如 MasterCard/VISA/运通等，提供金融交易的国际对接服务，也可以作为移动支付的服务提供商和应用提供商，如 Mastercard 的 Paypass/Masterpass、VISA 的 V. me 等。

⑦ 标准组织：制定移动支付相关标准的组织，如央行、工信部等。另外，如国外的 EMVco [EMV 标准是由国际三大银行卡组织：Europay（已被万事达收购）、MasterCard 和 Visa 共同发起制定的银行卡从磁条卡向智能 IC 卡转移的技术标准，是基于 IC 卡的金融支付标准，已成为公认的全球统一标准]、ISO 制定的 ISO 14443 等非接触智能卡标准，以及 GP（Global Platform，全球平台组织）开发和制定的智能卡相关技术标准，如 GP2. 2 标准。

⑧ 芯片厂商：移动终端上完成移动支付功能所需的硬件芯片。NFC 芯片厂商有恩智浦、意法半导体、博通等公司。

⑨ 服务提供商：提供移动支付服务，如中国移动、中国电信、银联、公交运营机构、公共收费机构等，还包括第三方支付方，如支付宝等。

⑩ 卡商：移动支付中的安全单元以 SIM 卡、SD 卡等智能卡形式存在。安全单元用于存放移动支付中与用户安全相关的数据，如用户账户、密钥等。卡商主要有金雅拓、捷德、东信和平、国民技术等公司。

⑪ 央行：制定移动支付标准，定义金融交易流程等，同时对移动支付产业的发展进行政策性引导。

⑫ 工信部：制定移动支付标准，同时对移动支付产业的发展进行政策性引导。

⑬应用提供商：移动支付应用的开发商，可以由服务提供商自己开发，也可以是第三方应用提供商来开发。

⑭ TSM 提供商：TSM 是负责移动支付安全单元管理及应用生命周期管理的平台。TSM 提供商一般为运营商或银行，也可以是手机厂商或应用服务提供商。

⑮ 测试认证组织：一般由银行、运营商等组成，制定移动支付交易流程、通信流程及安全、性能等方面的测试规范，负责移动支付终端的相关测试及认证工作。

虽然移动支付生态圈有着众多的参与方及不同的利益体，但是移动支付的相关产业链已形成。图 14-3 描述了生态链中各参与者之间的关系。

用户包括消费者和商家。消费者通过手机上的支付应用来使用服务提供商提供的移动支付服务，支付应用运行在安全单元中。商家通过 POS 机与消费者进行移动支付交易。服务提供商/应用提供商将自己的服务/应用发布在 TSM 上，用户通过无线网络，如 OTA（Over The Air，空中下载）方式下载或更新手机安全单元中的支付应用。

14.1.2 国内移动支付的发展

移动支付技术从诞生、发展至今已经有 10 年多的历史，但是在中国的发展速度比较缓慢，一直未得到大规模推广。一方面是由于移动终端能力的限制，移动互联网基础设施薄弱；另一方面，因为移动支付产业链不成熟，生态圈中各利益方无法达成一致。但是在最近两年，随着智能手机的普及，移动支付市场逐渐成熟。对许多消费者来说，手机可以代替现金、银行卡、公交卡等，携带起来非常方便，消费过程更加便捷、简单。因此，移动支付受

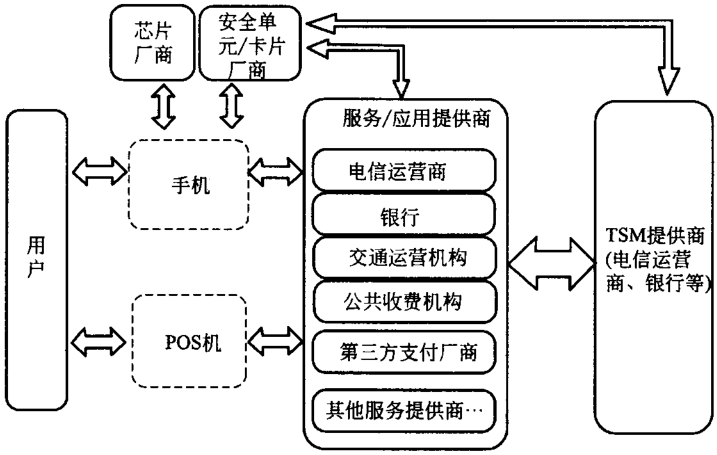


图 14-3 移动支付生态链

到运营商、支付厂商及银行的重视，他们纷纷投入到移动支付的开发和运营中来。在国内，随着移动支付的发展，出现了种类繁多的移动支付的相关应用产品。我国的移动支付技术经历了四个阶段的发展[1]。

第一阶段是通过手机短信的方式进行支付。它主要是将用户的手机号和用户的支付账号关联，完成支付过程。通常情况下，使用用户通信话费进行支付。2000 年 5 月，中国移动正式推出短信服务，并被很多服务提供商应用于手机代扣费的信息服务。短信支付被广泛应用在互联网和移动互联网的小额收费业务，如电子书刊、会员费缴纳、付费下载等。但是短信支付存在巨大的隐患和风险，支付环节不透明，话费计算混乱、不准确，使得用户蒙受了损失，移动运营商的声誉受到严重的负面影响。2006 年，工信部下发文件，不允许任何服务提供商提供任何形式的扣费业务，并设立扣费前用户二次确认等机制，使得手机话费作为支付资金来源受到较为严格的限制，手机话费短信支付从此步入低谷。

第二阶段是基于 WAP 的方式，利用移动终端的客户端或 WAP 浏览器，通过 GPRS 或 CDMA2000 1x 网络进行支付。这种方案既可以采用后台账户绑定模式，也可以采用在支付过程中录入账户信息的模式，如让用户输入银行卡号和密码。这种移动支付模式与第一代移动支付有同样的缺点，那就是安全性问题，另外还受到网络速度的制约。

第三阶段是一种非接触式移动支付模式。主流的非接触移动支付模式有 NFC、SIMPASS 以及 RF-SIM 三种比较成熟的技术。NFC 和 SIMPASS 使用 14.56MHz 频率，该频率和协议广泛应用在交通、金融等多个行业，是世界通行的标准。RF-SIM 技术是将包括天线在内的 RFID 射频模块与传统 SIM 卡功能集成在一张 SIM 卡上，在实现普通 SIM 卡功能的同时，通过射频模块完成各种移动支付。RF-SIM 卡是一种带有射频功能的安全 SIM 卡，如图 14-4 所示，其内部集成金融级的安全芯片和完全独立的射频通道。该射频通道工作在 2.45GHz 的频率上，通信距离达到 5m。SIM 卡部分用于正常的手机移动通信、鉴权，与手机物理连接。RF-SIM 卡广泛应用于企业门禁、考勤、食堂一卡通系统、交通等公共事业领域。目前 RF-SIM 方案的主要卡商有国民技术、中科讯联等。中国移动、中国联通、中国电信三家运营商曾在全国进行大面积试点，其中校园一卡通、企业一卡通等应用颇为广泛[2]。

---

[1] 中国移动支付行业研究报告．艾瑞咨询集团，2011．

[2] RF-SIM 卡．百度百科．

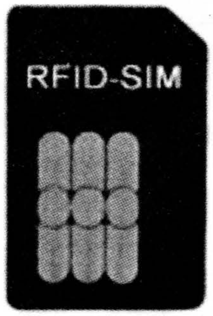


图 14-4 RF-SIM 卡

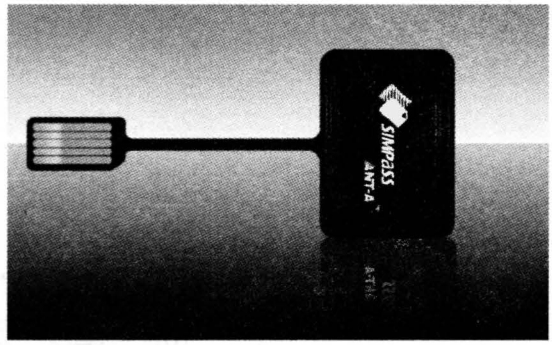


图 14-5 SIMPASS 卡

SIMPASS 采用双界面的智能 SIM 卡代替传统的 SIM 卡实现支付功能，如图 14-5 所示。SIMPASS 具有非接触和接触两个界面：接触界面是普通的 SIM 卡，而非接触界面可以支持各种非接触的金融应用。非接触界面符合 ISO-14443 规范，增加一个射频天线，便可进行非接触通信。天线与 SIMPASS 卡片的触点 C4 和 C8 连接，一般贴放在电池和手机后盖之间。SIMPASS 方案的提供商主要是握奇公司。

第四个阶段即 2013 年《中国金融移动支付系列技术标准》正式发布，规定将 14.56MHz NFC 技术作为我国移动支付近场支付的标准。该标准的确定让中国 NFC 支付商用进程在艰难前行的道路上迈出了重要一步。和国外相比，国内 NFC 支付起步并不算晚，但过程异常缓慢，这其中固然有国内金融环境的影响，但最主要的因素还是对于技术标准的争夺。

为了占领移动支付市场，三大运营商纷纷成立移动支付公司。中国移动于 2011 年 7 月成立中国移动电子商务有限公司，中国联通于 2011 年 4 月组建联通沃易付网络技术有限公司，中国电信 2012 年 3 月成立天翼电子商务有限公司。2011 年 12 月，三大运营商移动支付子公司同时获得央行颁布的支付业务许可证。另外，经过十多年的发展，我国第三方支付行业已形成较完备的产业链结构。截至 2013 年 3 月底，中国获得第三方支付牌照的企业数量已达 223 家[1]，涌现如支付宝、财付通、快钱等一大批第三方支付企业，这里不再详细介绍。

14.1.3 国外移动支付发展

日本是移动支付发展最好的国家之一，特别是基于 NFC 的移动支付。在日本，通过手机进行支付非常普遍，乘坐地铁、公交车、出租车等交通工具，商场购物，吃饭等日常消费，几乎都可以通过手机来实现。日本的 3 家运营商 NTT DoCoMo、KDDI、软银分别推出了移动支付业务，采用索尼公司的 Felica 技术。

日本市场的手机几乎全由运营商定制及出售，因此运营商具备决定手机硬件支持移动支付的能力。NTT DoCoMo 在入股瑞穗与三井住友两家银行之后，在移动支付领域中实现了产业链的完全覆盖。NTT DoCoMo 的移动支付以“Osaifu-keitai”为业务品牌，包含手机钱包、ID 借记卡与 DCMX 信用卡等三大类业务。其中，手机钱包服务主要采用 Felica 技术，与 NFC 兼容[2]。

此外，为了降低更新 POS 终端对零售商带来的成本冲击，NTT DoCoMo 选择了已经支持 NFC 移动支付的连锁便利商店 AM/PM、麦当劳、7-11 以及 JR-East（东日本铁路公司）

[1] 中国第三方支付行业发展研究报告简版. 艾瑞咨询集团，2012—2013.

[2] 舒杰，苏晟恺. NFC 移动支付是一场持久战.

为合作伙伴，并搭配降低交易费用的方式，普及移动支付。尤其是与 JR-East 达成合作，不仅极大地增加了支付网点，更帮助用户培养了移动支付习惯。目前，NTT DoCoMo、韩国 KT 以及中国移动正在合作推广跨国 NFC 移动支付漫游服务，允许三大运营商的用户使用当地的移动支付服务。

在美国，目前主流的移动支付方式有 Square、谷歌钱包（Google Wallet）、ISIS 等。这里先介绍 Square，后面将详细分析谷歌钱包和 ISIS。

Square 是于 2009 年 12 月在美国旧金山市成立的移动支付创业公司，主要解决个人和企业的移动终端支付问题。在国内，有众多仿效 Square 的产品，如盒子支付、钱袋宝、乐刷等。目前，Square 有 3 个部分：Square reader 读卡器、Pay with Square 便捷移动支付及 Square Register[1]。

通过手机的音频接口插入 Square 读卡器，手机就变成了一个小型 POS 机，如图 14-6 所示。商户和个人都可以免费申请使用，且付款、收款过程简单、高效。

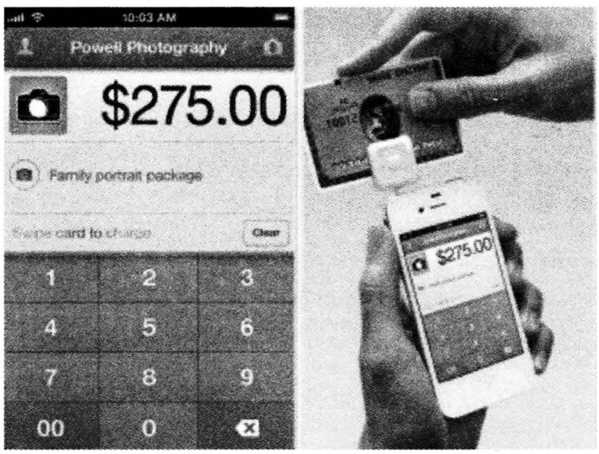


图 14-6 Square 读卡器

Square Register 目前只在 iPad 上供商户使用，功能类似收银台，需要配合读卡器使用，如图 14-7 所示。这不仅便利了商户和消费者的支付过程，更为商户进行客户维护和数据分析提供了可能。

Pay with Square 无需配合读卡器使用，是一个移动支付服务工具，只需店员根据消费者姓名确认个人身份后便可完成支付，如图 14-8 所示。其主要特点在于充分利用了移动设备的位置性、社交性和身份识别性，进一步简化了用户的支付过程，同时加入了社交、店铺管理、位置推送等元素。


图 14-7 Square Register

2012 年，Square 获得了 2500 万美元的投资，并与星巴克结为伙伴关系，而第四轮融资公司估值达到了 32.5 亿美元。美国 7000 家星巴克连锁店开始接受 Square 支付。截至 2012 年 9 月，Square 的年度交易量达到 80 亿美金。

❶ 海外移动互联网案例研究报告——Square. 艾瑞咨询集团，2012.



图 14-8 Pay with Square

14.2 NFC 移动支付技术

14.2.1 NFC 移动支付系统架构

央行颁布的中国金融移动支付的标准中定义了两种移动支付方式，即近场支付和远程支付。其中，近场支付即采用 NFC 的方式进行非接触支付；远程支付即移动终端通过移动网络，如 3G 网络，接入远程支付系统中进行支付。图 14-9 描述了 NFC 移动支付的系统架构。

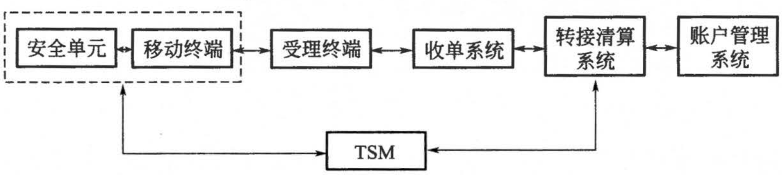


图 14-9 NFC 移动支付系统架构

① 移动终端：指在移动支付业务使用过程中，用户在移动状态下使用的终端设备，如手机等。

② 受理终端：指参与移动支付交易的交易专用终端设备，如 POS 机、ATM 机等。

③ 收单系统：负责联机交易信息的产生和转接，以及结算数据的收集、整理和提交等。

④ 转接清算系统：指实现跨机构支付业务转接、清算和结算功能的系统。

⑤ 账户管理系统：指为银行卡账户或非银行卡结算账户提供资金管理、结算等业务的系统。

⑥ TSM：负责移动支付安全单元管理及生命周期管理的实体。

⑦ 安全单元：用于安全敏感数据的存储，如对交易的关键数据进行安全存储和运算，确保敏感交易具有安全认证和不可抵赖性。安全单元以 SIM 卡、SD 卡或单芯片的形式存在。

根据安全单元与移动终端之间的连接关系，有三种技术解决方案：SD 卡方案、嵌入式全终端方案以及 SWP SIM 卡方案。这三种不同的方案实际代表着移动支付各个参与方的利益诉求，下面将详细介绍。

14.2.2 SD 卡方案

SD 卡方案是指将安全单元集成在 SD 卡中的方案，如图 14-10 所示。在传统的 SD 卡内嵌入安全单元芯片之后，形成用于金融交易的智能 SD 卡。

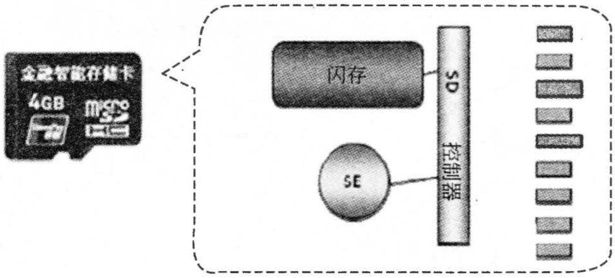


图 14-10　包含安全单元的金融 SD 卡

SD 卡中的安全单元与 CLF（Contactless Front，非接触前端）相连。这里 CLF 特指实现 NFC 功能的 NFC 芯片，如图 14-11 所示。当进行移动支付时，运行在安全单元中的移动支付应用通过 CLF 与受理终端，如 POS 机或 ATM 机进行 NFC 通信，从而完成金融交易。该方案需要修改手机硬件 SD 卡的相关电路，才能支持 CLF 与 SD 卡之间的通信。

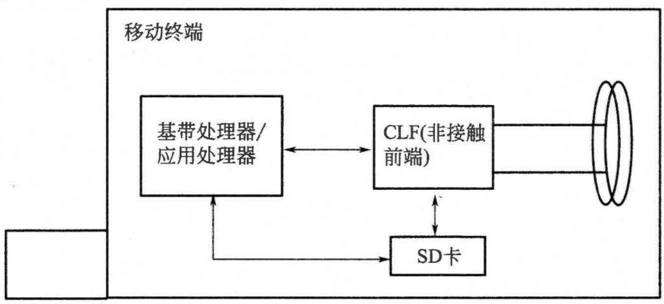


图 14-11　SD 卡方案

该方案的主要推动方是银联及各大银行。该方案需要对 SD 卡定制，并且需要改造手机硬件电路，没有得到手机厂商的广泛支持及市场认可，目前只有少数手机厂商支持，如 HTC、天语等。采用该方案的 SD 卡发行量在 5000 张左右。

14.2.3 嵌入式全终端方案

嵌入式全终端方案指将安全单元集成在手机的主板上，或者与 CLF 芯片集成在一个 SOC（System On Chip）芯片中，如图 14-12 所示。安全单元芯片的厂商主要是意法半导体，比如意法半导体公司推出的 ST33F1M 安全芯片，搭载了金雅拓公司开发的安全操作系统。该操作系统基于 Global Platform GP 2.2/Java 3.0.1 平台，是目前市场上最先进的 NFC 安全支付平台之一。另外，意法半导体公司推出了集成安全芯片的 NFC 芯片 ST33。

对于嵌入式全终端方案，安全单元芯片集成在手机电路板上，或集成在 NFC 芯片中，所以手机厂商掌握了对安全单元的控制权，给手机厂商提供了更多参与移动支付的机会。手机厂商可以建立自己的 TSM，或与其他业务提供商合作来建立 TSM 对安全单元的管理，从而在移动支付产业链中获取更大的利益。例如，三星公司新发布的智能手机同时支持 SWP SIM 和嵌入式全终端方案。三星公司也建立了自己的 TSM，并推出了“三星钱包”移动支付业务。

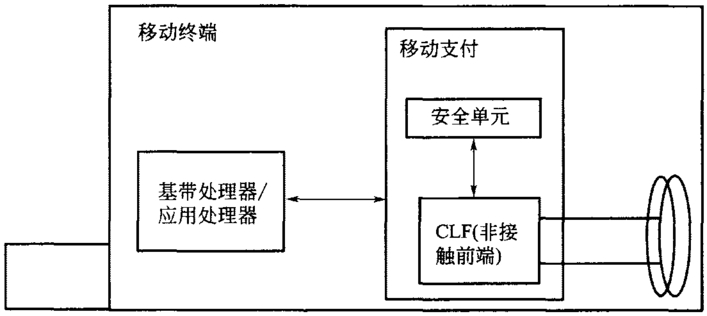


图 14-12　嵌入式全终端方案

14.2.4　SWP SIM 卡方案

SWP SIM 卡方案是指将 SIM 卡作为安全单元，通过 SWP（Single Wired Protocol）协议与 CLF 芯片连接的方案，如图 14-13 所示。

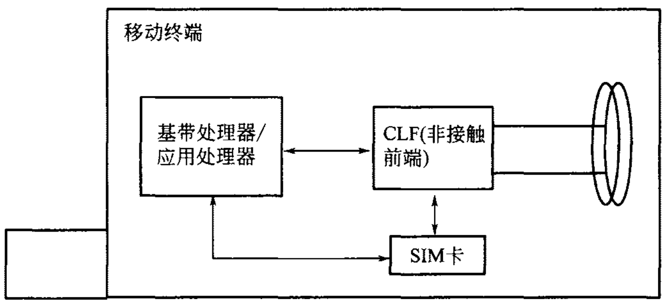


图 14-13　SWP SIM 卡方案

SWP 即 ETSI TS 163 规范，是一种单线协议，规定了 SIM 卡和 NFC 芯片之间的通信接口。该规范在一条单线上实现全双工通信，即 S1 和 S2 这两个方向的信号，如图 14-14 所示。通信的双方是 UICC（Universal Integrated Circuit Card，通用集成芯片卡）和 CLF。这里 SIM 卡是 UICC 的表现形式。UICC 还可以是 USIM（Universal Subscriber Identity Module）、ISIM（IP Multimedia Service Identity Module）等。为了方便起见，本书以 SIM 卡代替。S1 是电压信号，SIM 卡通过电压表检测 S1 信号的高、低电平，采用电平宽度调制；S2 是电流信号，采用负载调制方式。S2 信号必须在 S1 信号为高电平时才有效；S1 信号为高电平时，导通其内部的一个三极管，S2 信号才可以传输。S1 信号和 S2 信号叠加在一起，在一条单线上实现全双工通信。

SWP 是关于物理层和数据链路层的协议。物理层负责 SIM 卡和 CLF 之间物理链路的激活、保持、解除工作。数据链路层采用 ACT 协议和 SHDLC 协议。ACT 协议是接口激活协议，用于激活 SWP 接口。在没有外部射频场的情况下，SWP 接口处于去激活状态。在卡模拟式模式下，当感应到射频场后，NFC 芯片将被激活。SIM 卡收到 NFC 芯片的高电平信号后，使用 ACT 帧建立物理链路的连接。SHDLC 协议是高级数据链路控制规范的简单版本，主要用来传输交互的数据信息，其信息帧承载上层 HCP（主机控制协议）数据包。此外，SHDLC 协议还具有流控、错误检查、出错后数据重传等功能。

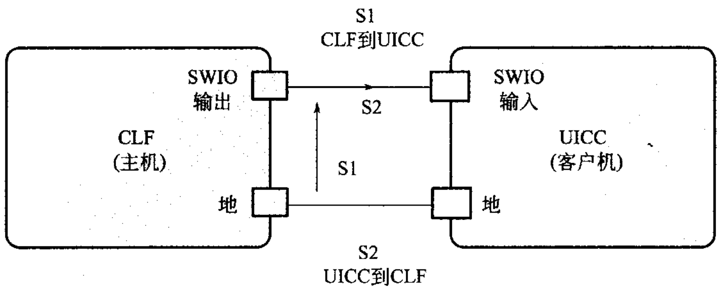


图 14-14 SWP 原理

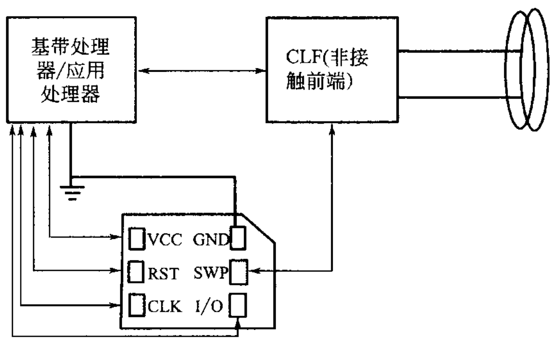


图 14-15 C6 管脚连接图

SIM 卡芯片一般有 8 个触点与手机相连：电源 VCC（触点 C1）、复位 RST（触点 C2）、时钟 CLK（触点 C3）、未使用（触点 C4）、接地端 GND（触点 C5）、编程电压 VPP（触点 C6）、数据 I/O 口（触点 C7）、未使用（触点 C8）。在这 8 个触点中，触点 4 和触点 8 一般不使用；与 CLF 进行 SWP 连接的管脚采用 C6 触点，如图 14-15 所示。

SWP 是物理层和数据链路层协议；HCI 定义了主机之间的接口，包括主机（Host）、端口（Gate）和管道（Pipe）逻辑协议，在 SWP 协议之上负责逻辑传输管道的建立和路由，如图 14-16 所示。HCI 对于底层的物理连接方式不做具体规定，可以是 SWP 协议，也可以是其他接口。

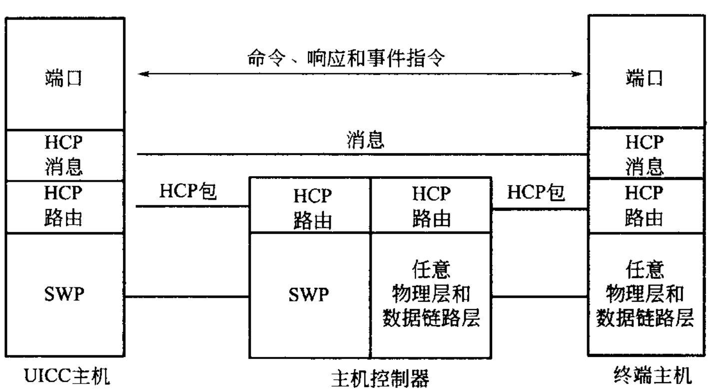


图 14-16 HCI 架构

主机是一个逻辑的概念，可以是终端主机、SIM 卡主机和主机控制器等。主机控制器是一种特殊的主机，用于管理主机间管道的建立、删除、维护等操作。端口是主机提供的服

务接入点，分为主机网络管理端口和通用端口。管道是两个端口之间传输信息的逻辑通道，分为静态管道和动态管道。静态管道无需创建和删除，而动态管道需要创建和删除。主机之间交互 HCP 数据包是通过主机控制器完成的。HCP 数据包中包含 PID（管道标识符），唯一地标识一个管道。主机控制器通过 PID 将 HCP 数据包转发给目的主机，目的主机再将该数据包交给相应的目的端口。HCI 规定了三类控制消息，即命令、响应和事件。命令用于管道的创建、删除、关闭等操作，响应用于回复命令的执行结果，事件用于通知主机某个事件的发生。

在 SWP SIM 卡的设计方案中，根据移动终端电池的情况，对 NFC 的三种工作模式有不同的要求。在正常情况下，即电池能够正常为移动终端供电，移动终端可以正常运行，要求对卡模拟、读写和点对点三种工作模式都支持；在低电情况下，即移动终端无法开机，但是电池还有余电，NFC 芯片只能工作在卡模拟模式；在无电情况下，即电池电压低于保护电压，NFC 芯片无法从电池获得电源供给，但仍然可以工作在卡模拟模式，在不损害电池的条件下，一般要求能够完成 20 次交易。另外可选的，NFC 芯片可以从读卡器的射频场获取电能，并提供给 SIM 卡，从而支持 SIM 卡参与的移动支付交易。这就要求 NFC 芯片内部的电源管理单元连接到 SIM 卡的 VCC 管脚，如图 14-17 所示。

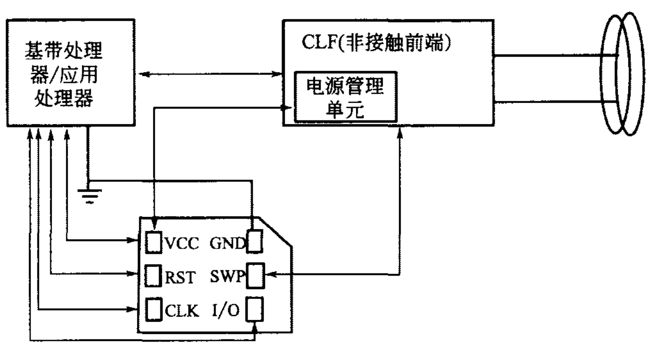


图 14-17 NFC 芯片为 SIM 卡供电

图 14-18 描述了移动终端的软件设计架构。

① NFC 支付应用：由应用或服务提供商提供的支付应用，如谷歌钱包、ISIS、一卡通应用等。

② NFC API：为上层应用提供访问 NFC 协议栈的 API 接口，如设置三种工作模式、RTD、NDEF 消息的解析等。

③ NFC 协议栈：实现了 NFC 三种工作模式及相关协议，对硬件调用提供统一接口，对操作系统资源进行封装抽象，为协议栈提供统一接口等。

④ SIM/SE API：为应用提供访问 SWP SIM 卡的接口，实现应用与 SWP SIM 之间 APDU 的交互，如 SIM Alliance 的 Open Mobile API。

⑤ RIL（Radio Interface Layer）：无线硬件与电话服务的抽象层，通过 AT 命令实现应用程序框架与基带芯片之间的通信。

⑥ 访问控制：提供对 SWP SIM 的安全访问管理，阻止应用对 SWP SIM 资源的非授权访问，保证业务安全。访问控制是通过存放在安全单元中的访问规则文件实现的。该文件定义了哪些应用可以访问安全单元中的哪些资源，或者哪些应用可以使用哪些 APDU 指令，如图 14-19 所示。这里，手机客户端就是指支付应用，访问控制模块在调用 SE API 时，需要先从安全单元中获取规则文件，并实现规则匹配的核心功能，如证书获取、规则管理、规

则匹配等。安全单元中的规则文件由 TSM 管理。

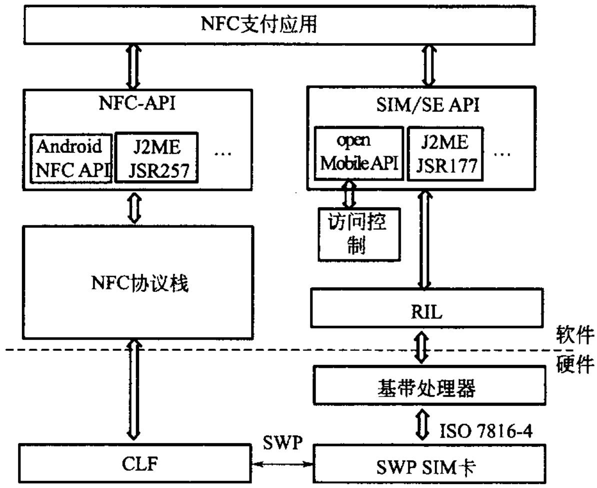


图 14-18 移动终端软件架构

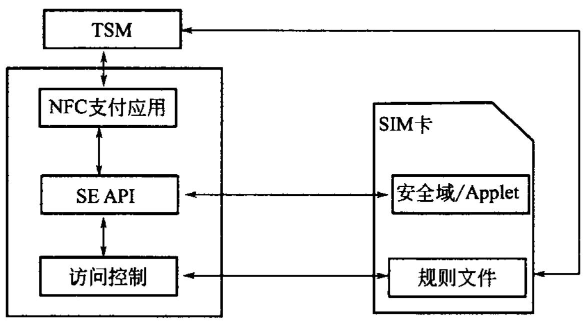


图 14-19 规则文件

14.2.5 NFC 移动支付流程

央行发布的《中国金融移动支付 近场支付应用 第 2 部分：交易模型及流程规范》中定义了 4 种交易模型：联机交易模型、脱机交易模型、移动终端交易模型和应用管理交易模型❶。

联机交易模型分为线下非接触 PBOC 交易模型和线下非接触 qPBOC 交易模型。这两种模型是指移动终端和受理终端分别采用 PBOC 标准和 qPBOC 标准来完成交易的过程。

脱机交易模型是指用户通过移动终端与受理终端交互进行支付，由受理终端直接承兑或拒绝，并在交易完成后通过收单系统和转接清算系统将交易通知或文件转发至账户管理系统进行清算处理。

移动终端交易模型是指交易只在移动终端和安全单元之间交互的交易模型。

应用管理交易模型是指用户使用移动终端通过客户端或应用管理终端完成如应用下载、应用删除、应用个人化和应用锁定等管理类交易模型。

本节主要针对线下 PBOC 交易、qPBOC 交易及脱机交易流程进行介绍。

---

❶ 《中国金融移动支付 近场支付应用 第 2 部分：交易模型及流程规范》.

14.2.5.1 线下 PBOC 交易流程

PBOC（The People's Bank of China，中国人民银行）规范又称为《中国金融集成电路（IC）卡规范》。该规范描述了电子钱包/存折应用，与 EMV 标准兼容的借/贷记应用，非接触式 IC 卡物理特性标准，电子钱包扩展应用指南，借/贷记应用个人化指南等内容。该规范最新的版本为 PBOC 3.0。

在线下非接触 PBOC 交易的处理流程中，安全单元与受理终端的交互流程如图 14-20 所示。

步骤 1：用户将移动终端放置于受理终端的射频场中，受理终端向安全单元发送命令，以获取发起交易所需的数据信息。

步骤 2：安全单元响应受理终端发起的命令，并将处理结果返回给受理终端。

步骤 3：受理终端向收单系统发送交易请求。

步骤 4：收单系统向受理终端返回交易应答。

步骤 5：受理终端向安全单元发送外部认证命令，安全单元响应外部认证命令，进行发卡行认证。

步骤 6：如果受理终端交易应答中存在脚本，安全单元执行脚本，并将脚本执行结果反馈给受理终端。

步骤 7：客户端向用户提示本次交易信息（此步骤可选）。

步骤 8：受理终端将发卡行脚本的执行结果反馈给收单系统。

步骤 9：收单系统向受理终端应答脚本执行结果通知。

上述流程适用于非接触 PBOC 方式下完成的 POS/ATM 联机交易，包括余额查询、消费、预授权等交易过程。

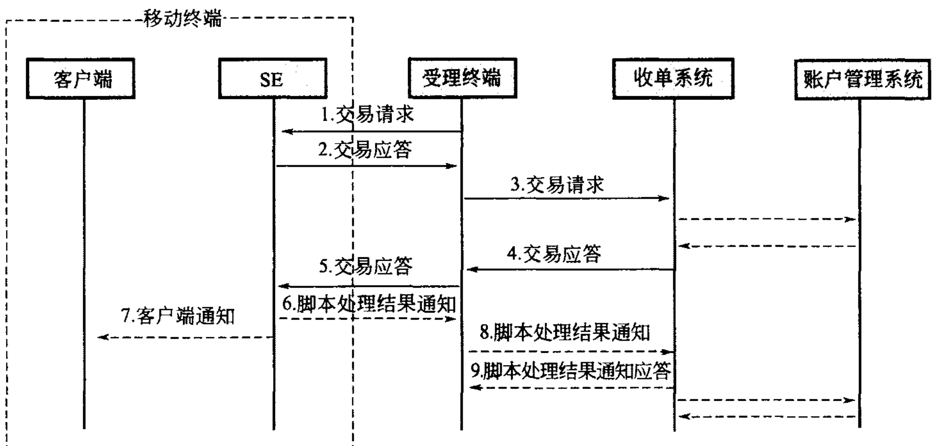


图 14-20 线下非接触 PBOC 交易处理流程

14.2.5.2 线下 qPBOC 交易

qPBOC 又称为快速 PBOC，定义了快速借记/贷记非接触式支付应用，以保证非接触快速交易。qPBOC 对标准的借记/贷记应用流程进行调整和优化，主要体现在：把多条命令压缩成尽可能少的命令，以减少交易的时间；将卡片和受理终端的交互过程集中完成，当卡片离开受理终端的通信范围后，受理终端再进行脱机数据认证、终端风险管理和终端行为分析，并允许卡片离开受理终端射频场感应范围之前或之后进行密码操作，使卡片在受理终端射频场感应范围停留的时间尽可能短。

在线下 qPBOC 交易的处理流程中，安全单元与受理终端的交互过程中涉及的详细步骤

如图 14-21 所示。

步骤 1：用户将移动终端放置于受理终端射频工作场（直至第 2 步结束后方可移出），受理终端向安全单元发送命令，以获取发起交易所需的数据信息。

步骤 2：安全单元响应受理终端发起的命令，并将处理结果返回受理终端。

步骤 3：客户端向用户提示本次交易信息（此步骤可选）。

步骤 4：受理终端向收单系统发起交易请求，收单系统请求账户管理系统获取交易结果。

步骤 5：收单系统向受理终端返回交易结果，受理终端显示交易结果。

上述流程适用于 qPBOC 方式完成的线下联机交易，包括余额查询、转账、消费、预授权等。

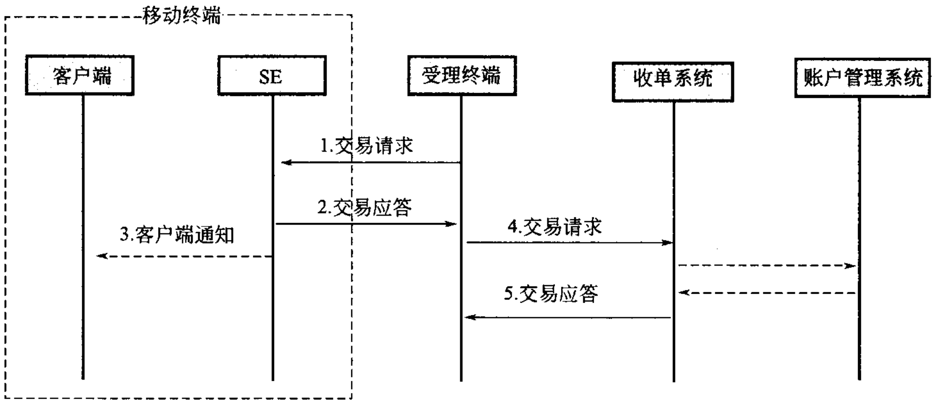


图 14-21 线下 qPBOC 交易处理流程

14.2.5.3 脱机交易

在脱机交易的处理流程中，安全单元与受理终端的交互过程中涉及的步骤如图 14-22 所示。

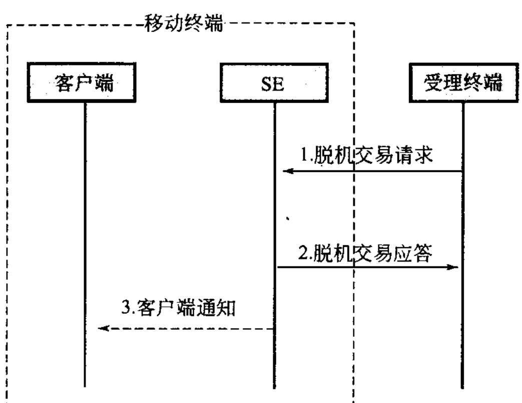


图 14-22 脱机交易处理流程

步骤 1：用户将移动终端放置于受理终端射频场内（直至第 2 步结束后方可移出），受理终端发起脱机消费交易。

步骤 2：安全单元响应受理终端发起的命令，并将脱机交易处理结果返回受理终端。

步骤 3：客户端向用户提示本次交易信息（此步骤可选）。

受理终端在同一清算日向收单系统上传清算数据。

上述流程适用于非接触 PBOC 或 qPBOC 方式完成的脱机交易，包括脱机消费、脱机余额查询等交易。

14.3 NFC 移动支付应用分析

随着 NFC 移动支付业务的开展，出现了很多基于 NFC 移动支付的应用程序或服务。本节主要介绍两个典型的 NFC 移动支付应用：谷歌钱包和 ISIS。

14.3.1 谷歌钱包

谷歌钱包是谷歌公司于 2011 年推出的一款基于 NFC 的移动支付应用，它不仅支持信用卡支付，还支持购物卡、礼品卡等支付方式。用户使用安装谷歌钱包的手机贴近具有 NFC 功能且支持谷歌钱包的 POS 机，就可以完成支付，如图 14-23 所示。



图 14-23 谷歌钱包支付

谷歌钱包主要有以下四个功能。

(1) My Wallet（我的钱包）

“我的钱包”功能存储了用户的个人信用卡信息。在 2011 年刚推出时，谷歌钱包只支持万事达信用卡和谷歌的预付费卡。目前谷歌钱包已经支持 Visa 和运通卡。谷歌预付卡是谷歌公司推出的金融卡，用户可以向该卡充值消费，目的是让谷歌钱包间接地支持大部分信用卡。如图 14-24 所示，谷歌钱包还支持在线向其他账户转账的功能。比如，朋友聚餐 AA 制时，由一个人统一结账，之后其他人通过谷歌钱包向付账的人结算。

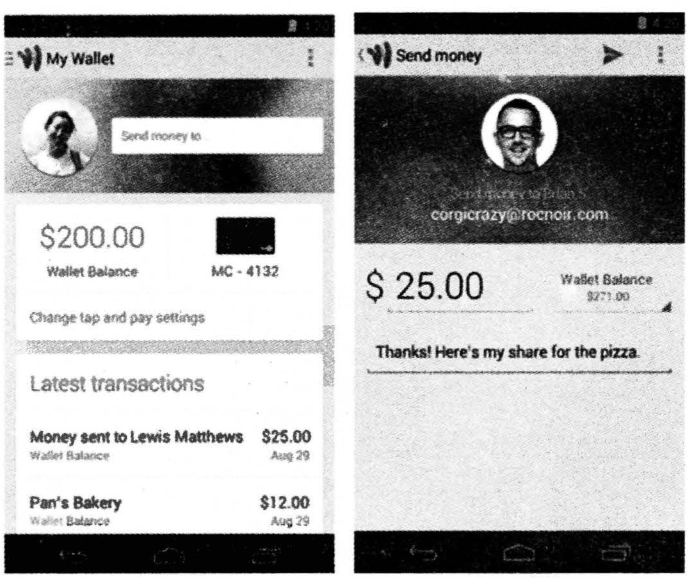


图 14-24 My Wallet 功能

（2）Loyalty Cards（会员卡）

“会员卡”功能存储了用户的会员卡信息，如图 14-25 所示。



图 14-25　Loyalty Cards 功能



图 14-26　My Offers 功能

（3）My Offers

谷歌钱包还集成了 Google Offer 功能，能够自动向用户递送团购优惠券信息，如图 14-26所示。用户只要在 My Offers 中选中一个优惠券，然后到商店中挑选相应的商品，再把手机放到读卡器前，就能完成支付。

（4）Transaction Details（交易记录详情）

交易记录详情用于显示最近支付的历史记录，如图 14-27 所示。

考虑到支付的安全性，谷歌钱包在手机黑屏状态下无法支付。另外，用户可以设置支付密码，在输入金额前，需要输入该密码。如果 30min 内没有使用钱包功能，再次刷卡时需要再次输入密码。使用谷歌钱包的智能手机一般采用嵌入式安全单元的方案。最初这个支付密码由谷歌钱包应用来维护，但是国外一家安全调查公司 ZVELO 发现，如果获得了手机的最高权限，谷歌钱包的支付密码将非常容易破解。因此，谷歌钱包最终将支付密码存放在安全芯片中，提高了其安全性。



图 14-27　Transaction Details 功能

谷歌钱包不仅展现了谷歌公司优秀的技术能力，而且展现了其超强的资源整合能力。谷歌公司甚至希望将来可以将驾照、医疗保险卡、护照等信息都集成在谷歌钱包中。目前谷歌钱包暂时没有制订任何收费计划，因此分析普遍认为谷歌钱包会通过谷歌公司的团购服务 Offers 来赚钱。

在移动支付领域，谷歌钱包不仅面对着像 Square、Paypal 这样实力强劲的竞争对手，而且要面对来自银行卡组织，如 Visa、万事达等各自推出的移动支付服务的挑战，就连美国运营商都联合起来推出自己的移动支付产品 ISIS，来对抗谷歌钱包。面对如此巨大的阻力，谷歌公司考虑开发支持 Android 平板的谷歌钱包产

品，让实体店售货员用它来读取信用卡。将跟 Square 形成直接的竞争关系。

14.3.2 ISIS

美国三大电信运营商 AT&T、T-Mobile、Verizon 于 2011 年 9 月 28 日合资成立 ISIS 公司，并推出了自己的 NFC 移动支付产品 ISIS，如图 14-28 所示。



图 14-28 ISIS 支付

ISIS 从一出世就被业界认为是谷歌钱包最有力的竞争对手。在 2011 年 Android 4.0 的 Galaxy Nexus 发布时，Verizon 就禁止在其上安装谷歌钱包应用，这表明 ISIS 和谷歌钱包是完全竞争的关系。2012 年，ISIS 在盐湖城推出试点项目，面向犹他州运输管理局下辖的公共交通系统推出一种替代信用卡和借记卡的支付方式。该项目还将支持盐湖城地区的部分零售网点。

ISIS 基本上与谷歌钱包的功能类似，可以导入并管理用户的银行信用卡、优惠券、会员卡等，如图 14-29 所示。

由于 ISIS 的主导者是运营商，因此 ISIS 采用了 SWP SIM 卡的方案，这是与谷歌钱包不同的地方。

通过对谷歌钱包和 ISIS 的分析可以看出，单从移动支付业务功能来说，主要是提供便捷的支付方式，导入并管理用户的银行信用卡、会员卡、优惠券等信息，给用户提供查询交易记录，并为用户提供支付的安全性保证等。从技术方案来说，由于移动支付本身牵涉到不同的利益方，因此不同的应用服务提供商会采用不同的技术方案，如嵌入式全终端方案或 SWP SIM 方案。就技术方案本身来说，并无优劣之分。移动支付的业务竞争更多地集中在资源整合和业务创新能力上。如何充分调动和整合移动支付产业链上的资源，发现参与方共赢的商业模式，充分挖掘消费者需求，并进行持续的业务创新，将是移动支付业务竞争的焦点。

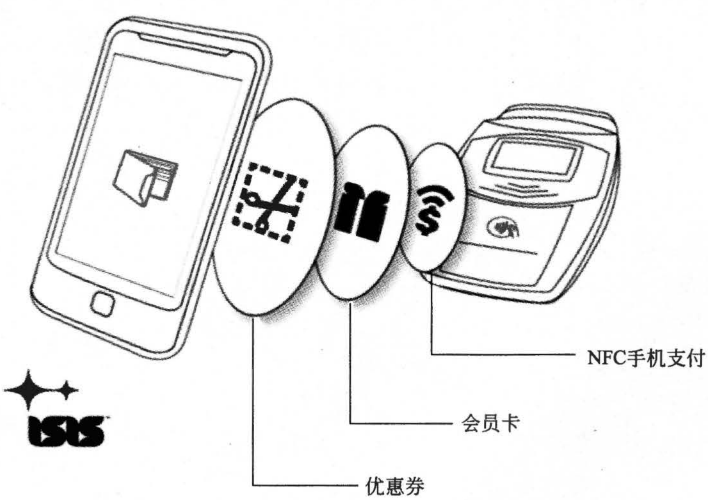


图 14-29 ISIS 功能

14.4 小结

本章介绍了移动支付的概念和分类，阐述了移动支付产业链中的各参与方。根据安全单元的不同，介绍了三种 NFC 移动支付技术方案：SD 卡方案、嵌入式全终端方案和 SWP SIM 卡方案，并详细介绍了 SWP SIM 卡方案的硬件架构及软件架构。同时，本章介绍了央行发布的移动支付标准中的非接触支付流程。最后分析了目前市面上的两种主流移动支付产品。

果粉藏書

参 考 文 献

[1] 高建良，贺建飚．物联网 RFID 原理与技术．北京：电子工业出版社，2013.

[2] 张新程．物联网关键技术．北京：人民邮电出版社，2011.

[3] 金光，江先亮．无线网络技术教程：原理、应用与仿真实验．北京：清华大学出版社，2012.

[4] Kevin R. Fall，W. Richard Stevens. TCP/IP 详解（卷 1：协议）．第 2 版。北京：机械工业出版社，2012.

[5] 中国科协学会学术部．无线电能传输关键技术问题与应用前景．北京：中国科学技术出版社，2012.

[6] 陈晓勤，钱守廉，李峰．移动支付改变生活：电信运营商的移动支付创新与实践．北京：人民邮电出版社，2012.

[7] 中国电信移动支付研究组．走进移动支付：开启物联网时代的商务之门．北京：电子工业出版社，2011.

NFC
技术原理与应用

销售分类建议：通信

ISBN 978-7-122-19983-6

www.cip.com.cn
读科技图书 上化工社网

定价：58.00元